

**VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA**

**Fakulta strojní**

**Katedra automatizační techniky a řízení**

*Aplikace neuronových sítí pro řízení laboratorních modelů*

Diplomant:

**Bc. Josef Voráček**

Vedoucí diplomové práce:

**Ing. Jolana Škutová, Ph. D.**

Ostrava 2010

## Anotace

**Voráček J,**     *Aplikace neuronových sítí pro řízení laboratorních modelů*  
Ostrava: Katedra automatizační techniky a řízení, Fakulta strojní  
VŠB – Technická univerzita Ostrava, 2010, 92s. Diplomová práce,  
vedoucí: Škutová J.

Diplomová práce se v první části věnuje teorii kolem neuronových sítí. Jedná se především o volby architektury neuronové sítě, algoritmy učení, inicializaci váhových parametrů a identifikaci pomocí neuronových sítí. Další část je věnována aplikaci neuronových sítí pro řízení laboratorního modelu stejnosměrného motoru. Byla provedena identifikace stejnosměrného motoru pomocí neuronových sítí a následná aplikace algoritmů řízení pomocí neuronových sítí na reálný model. Za tímto účelem byly vytvořeny simulační modely pro jednotlivé metody v prostředí Matlab/Simulink. Především byla použita metoda Přímého inverzního řízení, metoda Řízení s vnitřním modelem a Adaptivní regulace využívající ortogonální síť. Každá metoda je zhodnocena z hlediska kvality regulace integrálním kritériem IAE.

## Annotation

**Voráček J,**     *Neural Networks Application for Laboratory Models Control*  
Ostrava: Department of Control Systems and Instrumentation, Faculty of  
Mechanical Engineering  
VŠB – Technical University of Ostrava, 2010, 92p. Thesis,  
head: Škutová J.

Master thesis is in the first part deals with a theory about neural nets. This is essentially a choice of neural network architecture, learning algorithms, initialization parameters, weight and identification using neural networks. Another section is devoted to the application of neural networks for control of a laboratory model of a DC motor. Was carried out identification of DC motor using neural networks and subsequent application of control algorithms using neural networks to the real model. For this purpose, simulation models were developed for the different methods in Matlab/Simulink. Primarily used method of Inverse control method, Internal model control and Adaptive control using orthogonal network. Each method is evaluated in terms of quality control integral criterion IAE.

## Obsah

1	Úvod.....	6
2	Neuronová síť .....	7
2.1	Formální neuron.....	7
2.1.1	Vstupy neuronu.....	8
2.1.2	Váhy spojení .....	8
2.1.3	Práh neuronu .....	8
2.1.4	Agregační funkce .....	9
2.1.5	Aktivační funkce.....	9
2.2	Architektura neuronových sítí.....	11
2.2.1	Perceptronové sítě.....	11
2.2.2	Radiální síť .....	12
2.2.3	Rekurentní síť .....	13
2.2.4	Ortogonalní síť.....	16
2.3	Algoritmy učení neuronových sítí .....	17
2.3.1	Hebbův zákon učení.....	18
2.3.2	Chybová učení .....	18
2.4	Inicializace váhových parametrů .....	21
2.4.1	Náhodná inicializace .....	21
2.4.2	Nguyen – Windrow inicializace .....	22
2.4.3	Metoda inicializace založená na citlivostní analýze .....	22
3	Identifikace regulované soustavy s využitím neuronových sítí .....	25
3.1	Tréninková množina dat .....	27
3.2	Struktura modelu.....	27
4	Identifikace regulovaných soustav.....	32
4.1	Proporcionální soustava se setrvačností 3. řádu .....	32
4.2	Nelineární regulovaná soustava .....	39
5	Řízení regulované soustavy s využitím neuronových sítí.....	45
5.1	Přímé inverzní řízení pomocí neuronové sítě .....	45
5.1.1	Identifikace inverzního modelu regulované soustavy.....	46
5.1.2	Popis přímého inverzního řízení .....	46
5.1.3	Přímé inverzní řízení regulované soustavy se setrvačností 1. řádu .....	47
5.2	Řízení s vnitřním modelem.....	48
5.3	Dopředné řízení s inverzním modelem.....	49
6	Aplikace neuronových sítí na reálnou soustavu .....	51
6.1	Popis laboratorního modelu stejnosměrného motoru .....	51
6.2	Popis simulačního schématu reálné soustavy .....	52
6.3	Identifikace dopředného neuronového modelu stejnosměrného motoru .....	56
6.4	Přímé inverzní řízení stejnosměrného motoru .....	61
6.5	Řízení s vnitřním modelem stejnosměrného motoru .....	64
6.6	Adaptivní řízení .....	67
7	Závěr .....	74
	Seznam použité literatury .....	76
	Seznam příloh .....	78

## Seznam použitého značení

$a$	vstupní potenciál neuronu,
$d$	velikost predikce,
$e$	regulační odchylka
$e_{max}$	maximální regulační odchylka
$E$	kriteriální funkce,
$f$	aktivační funkce,
$f^0$	aktivační funkce ve výstupní vrstvě,
$f_i^h$	aktivační funkce $i$ -tého neuronu ve skryté vrstvě, kde $i = 1, 2, \dots, I$ ,
$\mathbf{g}$	vektor gradientů změny vah spojení,
$G(s)$	obrazový přenos,
$\mathbf{H}$	Hessova matice druhých derivací,
$\mathbf{I}$	jednotková matice,
$\mathbf{J}$	Jacobiova matice,
$k$	pořadí vzorku,
$k_p$	zesílení regulované soustavy,
$l$	počet neuronů ve skryté vrstvě,
$m$	počet minulých hodnot výstupní veličiny,
$n$	počet minulých hodnot vstupní veličiny,
$N$	konečný počet epoch trénování,
$o$	počet vstupů neuronové sítě,
$q^{-1}$	operátor zpětného posunutí,
$r$	maximální řád použitých ortogonálních funkcí
$s$	komplexní proměnná u obrazu L-transformace,
$T_l$	časová konstanta obrazového přenosu regulované soustavy,
$T$	vzorkovací perioda,
$t$	čas simulace,
$T_b$	vzorkovací perioda bloků v Matlabu,
$u_{IM}$	výstup inverzního modelu,

---

$u$	akční veličina,
$v$	porucha,
$\mathbf{W}$	matice vah spojení,
$w_0$	práh neuronu,
$w^0$	váhové spojení ve výstupní vrstvě,
$w_i^h$	váhové spojení $i$ -tého neuronu ve skryté vrstvě,
$\Delta w_{ij}$	změna vah spojení v čase $k$ mezi neuronem $i$ a neuronem $j$
$w$	žádaná hodnota,
$\mathbf{x}$	vektor vstupů neuronu, $\mathbf{x} = [x_1, \dots, x_i, \dots, x_I]^T$ ,
$y$	výstup ze soustavy,
$y_{NN}$	výstup z neuronové sítě,
$y_M$	obecný vektor výstupu modelu zastoupeného neuronovou sítí,
$y^0$	výstup neuronové sítě (také výstupní vrstvy),
$y_i^h$	výstup $i$ -tého neuronu ve skryté vrstvě,
$\alpha$	koeficient rychlosti učení,
$\beta_i$	časová konstanta $i$ -tého neuronu,
$\gamma_i$	vážený externí vstup,
$\varepsilon$	chyba predikce,
$\sigma(x)$	analytická nelinearita,
$\theta$	vektor parametrů,
$\kappa_r$	překmit žádané hodnoty,
$\lambda$	parametr rychlosti konvergence,
$\mu_{max}$	maximální hodnota z vlastních čísel autokorelační matice,
$\phi_{n_m}$	ortogonální funkce v $m$ -rozměrném prostoru,
$\varphi$	funkce závislosti výstupní veličiny na vstupní veličině regulované soustavy,
$\varphi^{-1}$	označení inverzní funkce.

## Seznam zkratek

<b>ARX</b>	autoregresní model s externím vstupem ( <i>Autoregressive with eXternal Input</i> ).
<b>ARMAX</b>	autoregresivní model s klouzavým průměrem a s dalším vstupem ( <i>AutoRegressive Moving Average with eXternal Input</i> ).
<b>BPG algoritmus</b>	algoritmus se zpětným šířením chyby ( <i>Backpropagation Gradient Descent Alorithm</i> ).
<b>FIR</b>	struktura modelu s konečnou impulsní odezvou ( <i>The Finite Impulse Response Model Structure</i> ).
<b>IAE</b>	integrální kritérium hodnotící kvalitu regulace na základě součtu absolutních hodnot regulační odchylky ( <i>Integral Absolute Error</i> ).
<b>IMC</b>	řízení s vnitřním modelem ( <i>Internal Model Control</i> ).
<b>IRC</b>	inkrementální rotační snímač ( <i>Incremental Rotary Sensor</i> ).
<b>LMS algoritmus</b>	algoritmus střední kvadratické chyby ( <i>Least Mean Square Algorithm</i> ).
<b>MSE</b>	střední kvadratická chyba ( <i>Mean Squarred Error</i> ).
<b>NARX</b>	nelineární autoregresní model s externím vstupem ( <i>Nonlinear Autoregressive with eXternal Input</i> ).
<b>NARMAX</b>	nelineární autoregresivní model s klouzavým průměrem a s dalším vstupem ( <i>Nonlinear AutoRegressive Moving Average with eXternal Input</i> ).
<b>NFIR</b>	nelineární struktura modelu s konečnou impulsní odezvou ( <i>Nonlinear Finite Impulse Response Model Structure</i> ).
<b>NOE</b>	nelineární model s chybou na výstupu ( <i>Nonlinear Error Model Structure</i> ).
<b>OE</b>	model s chybou na výstupu ( <i>Output Error Model Structure</i> ).
<b>PWM</b>	pulzní šířková modulace ( <i>Pulse Width Modulation</i> ).
<b>SSE</b>	součet kvadratických chyb ( <i>Sum Squared Error</i> ).

# 1 Úvod

Neuronové sítě se v historii svého vývoje vyskytly na obou pólech zájmu veřejnosti. Vlny jejich rozvoje i útlumu závisely především na technologických možnostech a rozvoji výpočetní techniky. Umělé neuronové sítě dokážou řešit komplikované problémy v oblasti identifikace a řízení až už lineárních tak zejména nelineárních regulovaných soustav. Jsou velkým příslibem pro tyto dvě oblasti řízení. Velkou výhodou umělých neuronových sítí je, že mohou pracovat s libovolným počtem vstupů a výstupů, proto jsou vhodné právě pro oblast identifikace a řízení vícerozměrných dynamických soustav.

Tato práce směřuje do oblasti aplikace neuronových sítí, popisu jejich architektury, inicializačních a učících algoritmů. Zabývá se taktéž možnostmi identifikace neuronovými sítěmi což je aplikováno na dvě regulované soustavy. Následně pak práce směřuje do oblasti popisu řídicích struktur pomocí neuronových sítí a je uveden jeden simulační příklad řízení pomocí neuronových sítí. Práce je taktéž zaměřena na oblast identifikace a řízení reálné soustavy, na kterou jsou aplikovány vybrané metody řízení. Jedná se o metodu Přímého inverzního řízení, metodu Řízení s vnitřním modelem a Adaptivní regulaci. Tyto metody jsou zhodnoceny z hlediska kvality regulace pro vybranou laboratorní úlohu. Všechny simulace a reálné řízení bylo provedeno v prostředí Matlab/Simulink, které podporuje aplikaci neuronových sítí.

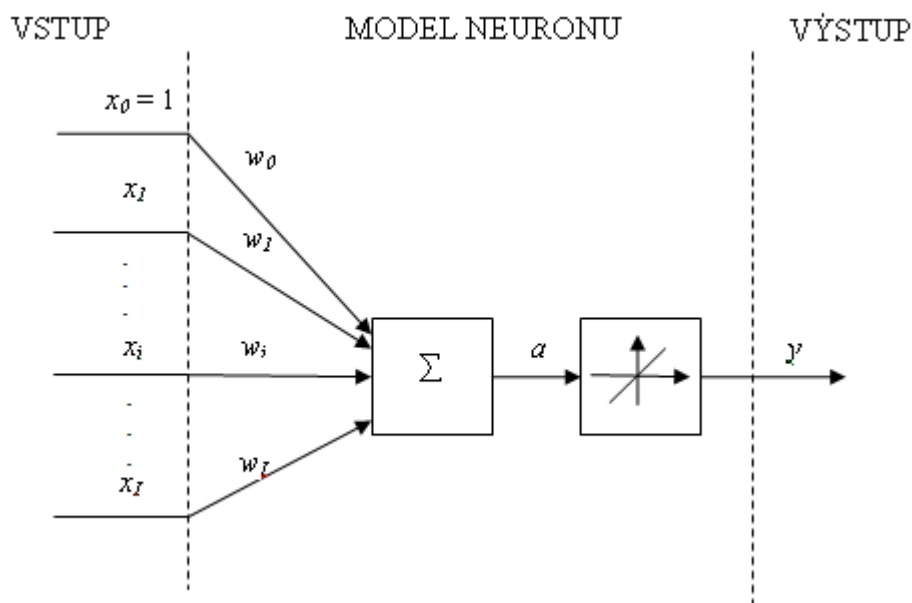
Využití umělých neuronových sítí je široké, je však nutno k nim přistupovat realisticky a vědět, že řadu problémů je vhodnější řešit jinými metodami.

## 2 Neuronová síť

Umělou neuronovou síť lze obecně charakterizovat jako nelineární stroj, který má schopnost se učit a je sestaven z prvků pro získávání informací. Každý prvek je spojen s jiným prvkem i sám se sebou pomocí zpětné vazby. Propojitelnost sítě definuje architektura neuronových sítí, která budou popsána v jedné z následujících kapitol [ŠKUTOVÁ, 2004].

### 2.1 Formální neuron

V umělém neuronu dochází k nahrazení biologických funkcí funkcemi matematickými. V praxi nejčastěji používaným modelem je formální model neuron. Často také nazývaný podle jmen autorů McCulloch-Pittsův neuron. [DRÁBEK, SEIDL, TAUFER, 2006]. Základní schéma umělého neuronu je na obr. 1.



Obr. 1 Model McCullochův-Pittsůva neuronu. [DOLEŽEL, 2008]

Formální neuron je popsán vstupy neuronu  $x_i$ , popř. vstupy z předcházející vrstvy umělé neuronové sítě, kde  $i$  je  $1, 2, \dots, I$  a  $w_i$  jsou synaptické váhy neuronu a prahová hodnota neuronu je zde značena  $w_0$ . Tyto hodnoty tvoří vstupy do agregační funkce. Výstupem agregační funkce (2.2) je vstupní potenciál  $a$  vstupující do aktivační funkce neuronu, jehož výstup je označen  $y$ .

Každý neuron obsahuje konečný počet vstupů  $x_i$  a jen jediný výstup  $y$ , který může být rozšířen o potřebný počet dalších neuronů. Tyto neurony tvoří pak vstup do následujících neuronů. Vstupní hodnoty se následně transformují na výstup pomocí minimálně dvou výpočetních procedur. První procedura je agregační funkce, jejíž



výstupem je potenciál neuronu  $a$  druhou je výpočet výstupu, který závisí na typu aktivační funkce  $f$ .

Na neuronovou síť je třeba nahlížet, jako na dynamický systém tzn., že jeho vlastnosti v průběhu trénování, pro adaptivní neuronové sítě pak i v čase mění [DRÁBEK, SEIDL, TAUFER, 2006].

### 2.1.1 Vstupy neuronu

Nejčastějším vstupem do neuronové sítě bývá vektor reálných čísel. Jedná se o skupinu vstupů, které jsou přivedeny a vstup z předcházejících neuronů.

Vstupní vektor  $\mathbf{x} = [x_1, \dots, x_n, \dots, x_l]^T$  pak tvoří soubor hodnot, které jsou buď „kvalitativní“ nebo „kvantitativní“. Kvalitativním vyjádření myslíme booleovské hodnoty jako *ano* nebo *ne*. Kvantitativní hodnoty tvoří skutečné hodnoty měřených veličin [DRÁBEK, SEIDL, TAUFER, 2006].

### 2.1.2 Váhy spojení

Jednotlivé vstupy jsou ohodnoceny konkrétní hodnotou příslušné váhy spojení. Tato hodnota reprezentuje citlivost, s jakou vstup působí na výstup neuronu. Provede se součet všech vstupů ohodnocených váhou spojení a výsledkem součtu je potenciál neuronu  $a$ . Váhy jsou nastaveny algoritmem učení neuronové sítě [DRÁBEK, SEIDL, TAUFER, 2006].

$$a_i = x_i \cdot w_i \quad (2.1)$$

### 2.1.3 Práh neuronu

Prahová hodnota znamená určitou hodnotu, kterou musí vstupní signál překonat, aby se mohl dále šířit neuronovou sítí. Hodnota prahu určuje, kdy je neuron aktivní nebo neaktivní. Je-li hodnota vstupního signálu nižší než hodnota prahová, výstupní signál z neuronu bude odpovídat pasivnímu stavu neuronu a naopak je-li hodnota vstupního signálu vyšší než hodnota prahová, výstupní signál odpovídá aktivnímu stavu neuronu. Prahové hodnoty se samozřejmě mohou v průběhu procesu měnit.

V modelech neuronových sítí se prahová hodnota používá k posunutí rozsahu hodnot do aktivační funkce. V tomto případě se prahová hodnota aktivně podílí na hodnotě výstupu z neuronu. Práh neuronu se modeluje jakou násobek synaptické váhy  $w_0$  a 1 resp. -1, kde číslice 1 a -1 představují fiktivní vstup do neuronu, podle toho, zda se signál šíří doprava resp. doleva [DRÁBEK, SEIDL, TAUFER, 2006].

### 2.1.4 Agregáčn  funkce

Klade si za c l ur it m zp sobem slou it vstupn  sign ly  $x_i$ , tzn. transformovat  $n+1$  – rozm rn  vstupn  vektor  $x_i (i = 0, 1, \dots, I)$  na skal rn  sign l  $a$ , kter  d le pokra uje na vstup aktiva n  funkce neuronu, kter  ur uje vztah mezi agrega n  funkc  a v stupem z neuronu. Nej ast ji pou  vanou agrega n  funkc  pro v stupn  vrstvu se pou  v  line rn  bazick  funkce pops n  vztahem (2.2). Pro skrytou vrstvu se pou  v  tangenci ln  aktiva n  funkce pops n  vztahem (2.3)

$$y = f(a) = w_0 \cdot a \quad (2.2)$$

$$y = f(a) = \tanh(a) = \frac{\sinh(a)}{\cosh(a)} = \frac{(e^a - e^{-a})}{(e^a + e^{-a})} = \frac{(e^{2a} - 1)}{(e^{2a} + 1)} \quad (2.3)$$

### 2.1.5 Aktiva n  funkce

Aktiva n  funkce je ur ena k p evodu skal rn  hodnoty  $a$  na skal rn  hodnotu  $y$ . Tvary aktiva n  funkc  jsou r zn . Lze je roz  lit na *line rn * a *neline rn *, *spojit * a *diskr tn *. V b r aktiva n  funkce je z visl  na p  padu ře en   lohy. Volba aktiva n  funkce ovliv uje n ro nost technick  pop . programov  realizace navr en  neuronov  s t . B  n  se pou  v  neline rn  aktiva n  funkce ve skryt  vrstv  a line rn  ve v stupn  vrstv .

Nej ast ji pou  van  aktiva n  funkce jsou

- Skokov  aktiva n  funkce.
- Aktiva n  funkce signum.
- Line rn  aktiva n  funkce.
- Sigmoid ln  aktiva n  funkce.
- Hyperbolicko-tangenci ln  aktiva n  funkce.

kde skokov  aktiva n  funkce (obr. 2) je vyj d ena n sleduj c m vztahem

$$y = f(a) = \begin{cases} 1 & a \geq 0 \\ 0 & a < 0 \end{cases} \quad (2.4)$$

Obdobnou aktiva n  funkc  je aktiva n  funkce signum (obr. 3). Lze j  popsat vztahem

$$y = f(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a < 0 \end{cases} \quad (2.5)$$

Následující vztah patří aktivační funkci, která se nazývá lineární aktivační funkce (Obr. 4)

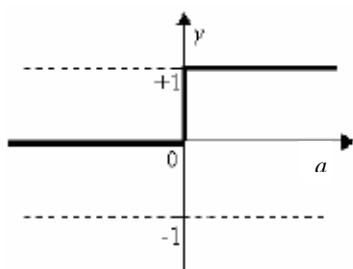
$$y = f(a) = w_0 \cdot a \quad (2.6)$$

Další aktivační funkcí je sigmoidální aktivační funkce (obr. 5)

$$y = f(a) = \frac{1}{1 + e^a} \quad (2.7)$$

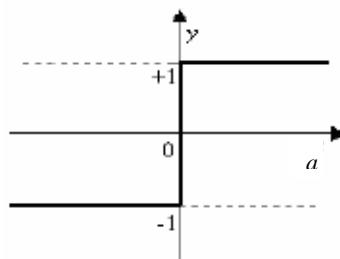
Poslední nejčastěji používanou aktivační funkcí je hyperbolicko-tangenciální aktivační funkce (obr. 6). Matematicky je jí možné popsat následujícím vztahem

$$y = f(a) = \tanh(a) = \frac{\sinh(a)}{\cosh(a)} = \frac{(e^a - e^{-a})}{(e^a + e^{-a})} = \frac{(e^{2a} - 1)}{(e^{2a} + 1)} \quad (2.8)$$



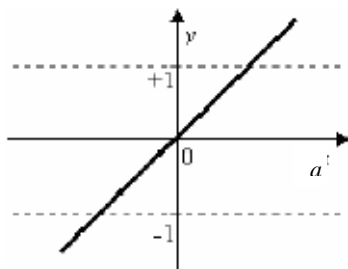
Obr. 2 Skoková aktivační funkce

[ŠKUTOVÁ, 2004]



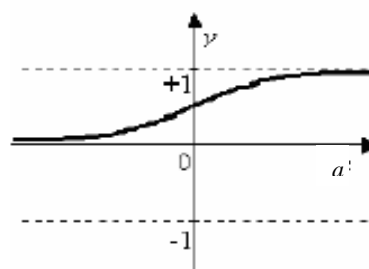
Obr. 3 Signum aktivační funkce

[ŠKUTOVÁ, 2004]



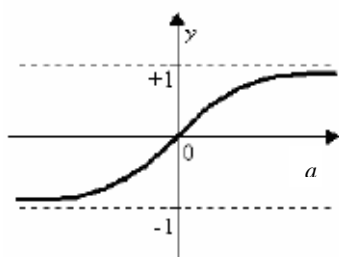
Obr. 4 Lineární aktivační funkce

[ŠKUTOVÁ, 2004]



Obr. 5 Sigmoidální aktivační funkce

[ŠKUTOVÁ, 2004]



Obr. 6 Hyperbolicko-tangenciální aktivační funkce [ŠKUTOVÁ, 2004]

Existuje ještě řada dalších aktivačních funkcí, např. saturační lineární aktivační funkce, symetrická saturační lineární aktivační funkce, aktivační funkce radiálního základu, aktivační funkce trojúhelníkového základu. Podrobně zde nejsou popsány, ale je možné je najít např. v [ŠKUTOVÁ, 2004].

## 2.2 Architektura neuronových sítí

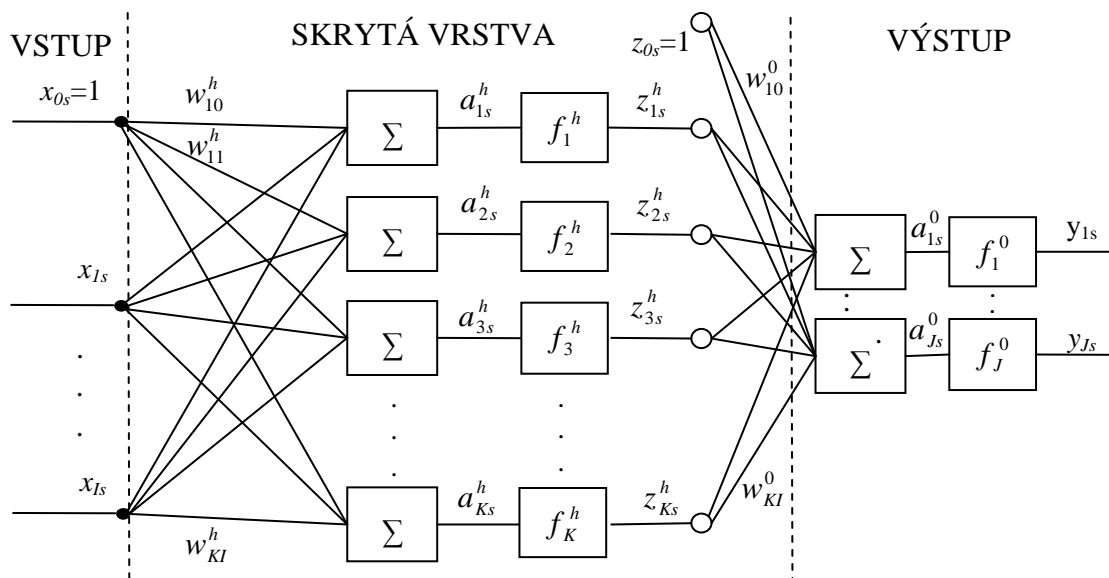
Architektura neuronových sítí představuje uspořádání neuronů a jejich vzájemné vazby. Volba architektury je jedním z nejdůležitějších kroků při návrhu neuronových sítí, která odpovídá typu řešené úlohy. Pro oblast identifikace a řízení systémů se nejčastěji využívají perceptronové sítě, které dále budou aplikovány v této práci [ŠKUTOVÁ, 2004].

V současnosti se využívají perceptronové, radiální, rekurentní, Hopfieldovy, Elmanovy a ortogonální sítě.

### 2.2.1 Perceptronové sítě

Perceptronové sítě mají dopřednou architekturu. Využívají pevného počtu vrstev a neuronů v těchto vrstvách. Počet vstupů se odvíjí od složitosti řešené problematiky [ŠKUTOVÁ, 2004].

Základní jednotkou této sítě je perceptron. Základním výkonným prvkem perceptronových sítí je model neuronu s lineární agregační funkcí a skokovou aktivační funkcí – jde o zvláštní případ základního McCulloch-Pittsova modelu neuronu znázorněného na obr. 1 [DRÁBEK, SEIDL, TAUFER, 2006]. Na obr. 7 je znázorněna vícevrstvá perceptronová síť.

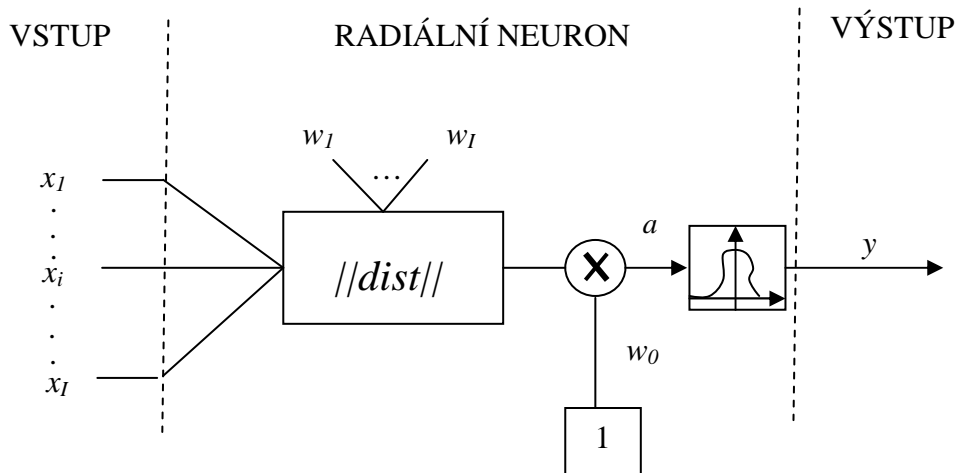


Obr. 7 Dvouvrstvá perceptronová síť

### 2.2.2 Radiální síť

Radiální sítě jsou alternativou perceptronových sítí. Mají vždy dvouvrstvou architekturu a počet neuronů závisí na algoritmu trénování. Tento počet je dán požadovanou přesností neuronové sítě a bývá mnohem vyšší než u perceptronových sítí. [ŠKUTOVÁ, 2004]

Základní jednotkou radiální sítě je radiální neuron viz obr. 8, kde  $x$  jsou vstupy,  $n$  je počet vstupů,  $w$  jsou váhy spojení,  $w_0$  je prahová hodnota (bias),  $f(a)$  aktivační funkce a  $y$  je výstup radiálního neuronu.



Obr. 8 Radiální neuronová síť[ŠKUTOVÁ, 2004]

Princip činnosti spočívá v tom, že vstup  $a$  radiální aktivační funkce radiálního základu je dán vektorovým rozdílem vektoru vah spojení a vstupů vynásobený prahem radiálního neuronu  $w_0$  [ŠKUTOVÁ, 2004]. Pak výstup radiálního neuronu je dán vztahem

$$y = f(\|w - x\|, w_0) = f(dist(w, x), w_0) , \quad (2.9)$$

kde následující vztah vyjadřuje rozdíl vektoru vah spojení  $w$  a vektoru vstupu  $x$

$$dist(w, x) = \sqrt{\sum_{i=1}^n (w_i - x_i)^2} . \quad (2.10)$$

Aktivační funkce je dána vztahem

$$y = f(dist(w, x)) = f(a) = e^{-(a)^2} \quad (2.11)$$

Aktivační funkce radiálního neuronu má maximální hodnotu na výstupu 1 a to tehdy, když vstup  $a$  je roven 0. Při snižování rozdílu mezi vektorem váhových hodnot  $w$  a vstupním vektorem  $x$  se hodnota výstupu neuronu zvyšuje [ŠKUTOVÁ, 2004].

### 2.2.3 Rekurentní síť

Architektura rekurentní sítě je vůči perceptronové síti rozšířená o zpětné vazby mezi jednotlivými vrstvami sítě. Vícevrstvá perceptronová síť představuje speciální případ rekurentní sítě, proto lze rekurentní síť pokládat za obecnou skupinu architektur [ŠKUTOVÁ, 2004].

Matematický popis rekurentní sítě se dvěma vrstvami je dán vztahem

$$y = f^0\left(\sum_{i=1}^q w_i^0 f_i^h\left(\sum_{j=1}^p w_{ij}^h x_j + \sum_{j=1}^p r_i^h f_i^h(\bullet, k-1) + w_i^h\right) + w^0\right) \quad (2.12)$$

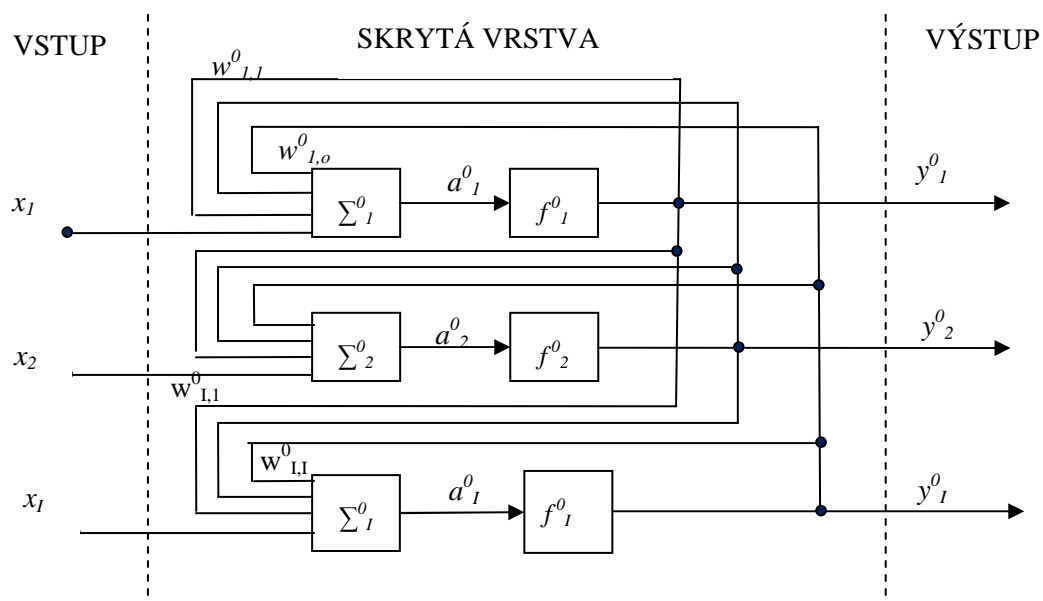
V dnešní době se nejčastěji využívají dva typy rekurentních sítí: Hopfieldova a Elmanova neuronová síť.

### Architektura Hopfieldovy sítě

Jedná se o neuronovou síť, kde výstupy neuronů jsou vedeny zpět na sebe samotné a taky ke všem ostatním neuronům. Tuto architekturu lze popsat diferenciálními rovnicemi, které následně tvoří množinu řešení [ŠKUTOVÁ, 2004].

$$\dot{x}_i = -\beta_i x_i + \sum_{j=1}^n w_{i,j} \sigma(x_j) + \gamma_i u \quad i = 1, 2, \dots, I \quad y = x_I \quad (2.13)$$

kde  $I$  je počet neuronů,  $x_i$  představuje stav  $i$ -tého neuronu,  $w_{i,j}$  je váha mezi  $i$ -tým a  $j$ -tým uzlem,  $\beta_i$  zahrnuje časovou konstantu  $i$ -tého neuronu,  $\gamma_i$  je vážený externí vstup,  $y$  je výstup sítě a  $\sigma(x)$  je analytická a pozitivní nelinearita, která se obvykle definuje pomocí *hyperbolicko-tangenciální* aktivační funkce a *sigmoidální* aktivační funkce [ŠKUTOVÁ, 2004].



Obr. 9 Hopfieldova síť [ŠKUTOVÁ, 2004]

Hopfieldova síť bývá testována jedním nebo více vstupními vektory a jsou předány jako počáteční podmínky do neuronové sítě. Následně síť produkuje výstupní hodnoty, které jsou zpětně předány na vstup. Proces probíhá tak dlouho, dokud se síť nestabilizuje [ŠKUTOVÁ, 2004].

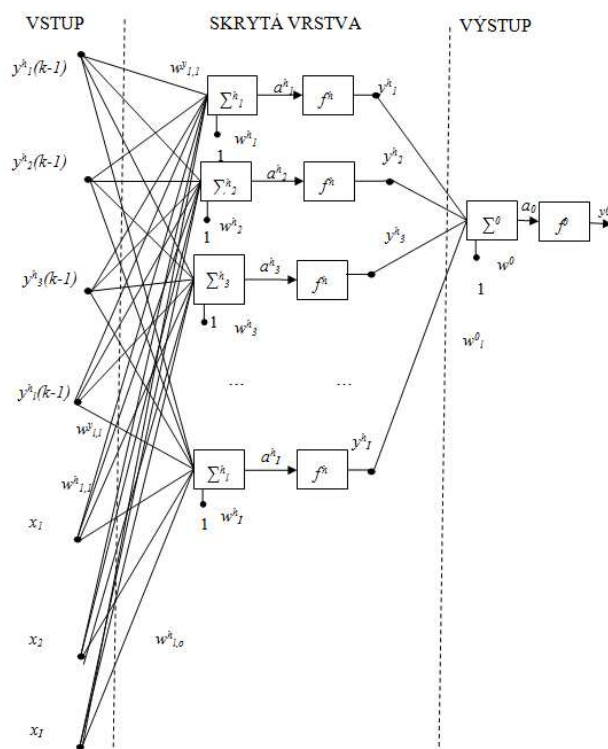
**Architektura Elmanovy sítě**

Opět se obvykle jedná o dvouvrstvou síť se zpětnou vazbou mezi výstupem a vstupem první vrstvy. Toto zpětovazebné spojení umožňuje produkovat časově-různé vzory [ŠKUTOVÁ, 2004]. Vztahy pro Elmanovu síť s jedním výstupním neuronem jsou dány

$$y_i^h(k) = f^h \left( \sum_{j=1}^r w_{i,j}^a y_j^h(k-1) + \sum_{k=1}^n w_{i,k}^h x_k(k-1) + w_i^h \right) \quad i = 1, 2, \dots, I \quad (2.14)$$

$$y^0(k) = f^0 \left( \sum_{i=1}^I w_i^0 y_i^h(k) + w^0 \right) \quad (2.15)$$

kde  $w_{ij}^0$  jsou váhy spojení mezi skrytými neurony v čase  $k-1$  a skrytými neurony v čase  $k$ ,  $y_j^k$  jsou skryté neurony v čase  $k-1$  (zpětná vazba od skryté vrstvy ke vstupní vrstvě),  $w_{i,k}^h$  jsou váhy spojení mezi vstupní a skrytou vrstvou,  $w_i^h$  a  $w^0$  jsou prahové hodnoty skryté a výstupní vrstvy  $f^h$  a  $f^0$  jsou aktivační funkce skryté (*hyperbolicko-tangenciální* nebo *sigmoidální*) a výstupní vrstvy (lineární funkce) [ŠKUTOVÁ, 2004].



Obr. 10 Elmanova síť [ŠKUTOVÁ, 2004]

Elmanova síť má tansigmoidální aktivační funkci v její skryté (rekurentní) vrstvě a lineární aktivační funkci ve výstupní vrstvě. Tato kombinace je jedinečná v tom, že



dvouvrstvá síť s těmito aktivačními funkcemi může aproximovat jakoukoli funkci (s konečným počtem nespojností) s libovolnou přesností. Jediným požadavkem je dostatečné množství neuronů ve skryté hladině, což vede ke složitosti sítě a její časové náročnosti na výpočet [ŠKUTOVÁ, 2004].

### 2.2.4 Ortogonální síť

Ortogonální síť je dopřednou neuronovou sítí s více vstupy a jedním výstupem, která využívá ve vnitřní skryté vrstvě ortogonální aktivační funkce. Vstupy ortogonální sítě se rozvětvují do bloků ortogonálních neuronů pro daný vstup. Tato neuronová síť s ortogonálními aktivačními funkcemi má dvouvrstvou strukturu. Skládá se ze vstupní, výstupní a skryté vrstvy. Skrytá vrstva se skládá z neuronů, které právě mají ortogonální aktivační funkce. Vstupní a výstupní vrstva se skládá z lineárních neuronů [KAJAN, 2006].

Uzly napravo od ortogonálních neuronů realizují součin ( $\Pi$ ). Každý uzel má  $m$  vstupních signálů z bloku ortogonálních neuronů. Tyto uzly jsou považované za část skryté vrstvy, přestože tam nejsou žádné váhové operace mezi ortogonálními neurony a  $\Pi$  uzly [KAJAN, 2006].

Výstup sítě je dán lineární kombinací aktivačních funkcí daných vztahem

$$\hat{y}(\mathbf{x}, \hat{\mathbf{w}}) = \sum_{n_1=0}^{(N_1-1)} \dots \sum_{n_m=0}^{(N_m-1)} \hat{\mathbf{w}}_{n_1 \dots n_m} \boldsymbol{\Phi}_{n_1 \dots n_m}(\mathbf{x}) = \boldsymbol{\Phi}^T(\mathbf{x}) \hat{\mathbf{w}} \quad (2.16)$$

kde  $\mathbf{x} = [x_1, x_2, \dots, x_I]^T$  je  $m$ -rozměrný vstupní vektor,  $N_i$  je počet neuronů pro  $i$ -tý vstup a  $\hat{\mathbf{w}}$  jsou váhy výstupní vrstvy. Funkce  $\boldsymbol{\Phi}_{n_1 \dots n_m}(\mathbf{x})$  jsou ortogonální funkce v  $m$ -rozměrném prostoru získané pomocného vztahu (2.17) [KAJAN, 2006].

$$\boldsymbol{\Phi}_{n_1 \dots n_m}(\mathbf{x}) = \prod_{i=1}^m \phi_{n_i}(x_i) \quad (2.17)$$

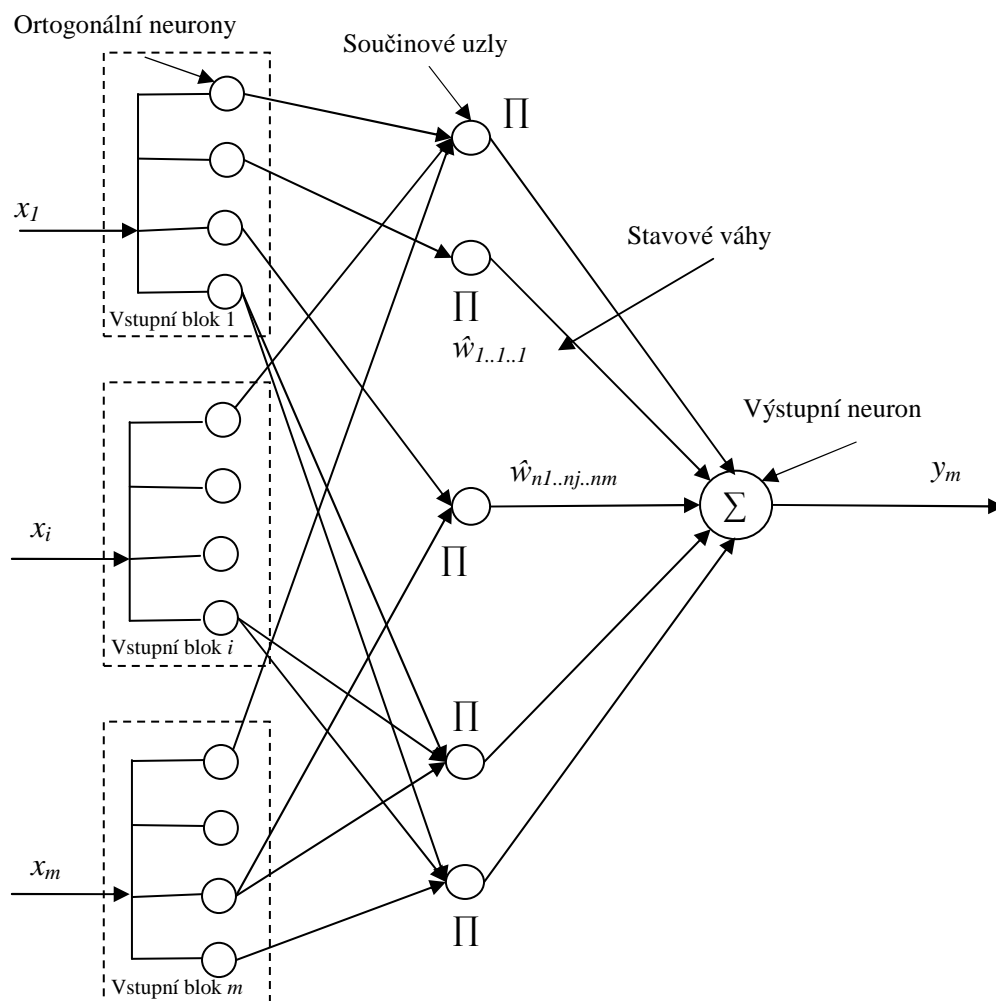
kde  $\phi_i$  jsou ortogonální funkce vložené do každého neuronu.  $\boldsymbol{\Phi}(\mathbf{x})$  a  $\hat{\mathbf{w}}$  transformují vstupní vektor a váhy, nebo-li  $n$ -dimenzionální vektor vah, kde  $N_{ortg}$  je vyjádřeno následujícím vztahem [KAJAN, 2006].

$$N_{ortg} = \prod_{i=1}^m N_i \quad (2.18)$$

Ze vztahu (2.19) je vidět, že roste počet váhových spojení s počtem vstupů neuronové sítě a řádem ortogonálních funkcí pro každý vstup. Platí naprostá propojitelnost.

$$i_1 + i_2 + \dots + i_m \leq r \text{ pro součin } \varphi_{i_1 1} \cdot \varphi_{i_2 2} \cdot \varphi_{i_m m} \quad (2.19)$$

kde  $r = \max (N_i - 1)$  a  $N_i$  je maximální řád použitých ortogonálních funkcí pro vstup  $i$ .



Obr. 11 Struktura neuronové sítě s ortogonálními aktivačními funkcemi [KAJAN, 2006]

## 2.3 Algoritmy učení neuronových sítí

Dalším důležitým krokem získání požadovaného modelu neuronovou sítí bývá volba algoritmů učení. Algoritmus učení nastavují parametry neuronové sítě tak, aby vznikla maximální shoda mezi modelem neuronové sítě a modelovanou regulovanou soustavou, tj. aby odchylka mezi skutečným a požadovaným výstupem z neuronové sítě byla minimální.

[ŠKUTOVÁ, 2004]

Algoritmus učení vyžaduje data pro nastavování vah spojení neuronové sítě. Podle typu modelované regulované soustavy je provedena volba architektury neuronové sítě, včetně počtu vrstev a neuronů v těchto vrstvách. Algoritmy učení lze rozdělit do dvou skupin přístupů a dvou skupin podle toho, zda jsou známy požadované výsledky či nikoliv.

Rozdělení algoritmů učení podle přístupu:

- **Skupinový přístup (dávkový)** – Tréninková data jsou dána již před začátkem trénování a nastavení parametrů probíhá ve všech krocích na základě známých vstupně-výstupních vzorů dat. Dále se metody pracující s tímto přístupem dělí na:
  - **Metody prvního řádu** – využívají pouze gradientní kritériální funkci.
  - **Metody druhého řádu** – využívají i derivaci kritériální funkce. Do této skupiny patří např. *Newtonova metoda*, *kvasi-Newtonova metoda*, *Gaussova-Newtonova metoda* a *Levenbergova-Marquardtova metoda*. Vyznačují se vysokou robustností a rychlou konvergencí.
- **Postupný přístup (interaktivní)** – Tréninková data jsou dodávána postupně a v každém kroku jsou dána pouze data aktuální pro daný krok. Tento přístup se využívá při modelování dynamického systému, pokud se jeho dynamika v čase mění [ŠKUTOVÁ, 2004].

V následujících kapitolách si popíšeme algoritmy učení a jejich modifikace.

### 2.3.1 Hebbův zákon učení

Hebbův zákon učení tvoří základ všech současných algoritmů učení. Každý neuron má pouze dva stavy, aktivní (1) a pasivní (0). Lze jej popsat vztahem

$$w_{ji}(k+1) = w_{ji}(k) + \alpha \cdot y_i(k) \cdot x_j(k) \quad (2.20)$$

kde  $w_{ji}$  jsou hodnoty vah spojení v čase  $k+1$ , jsou rovny hodnotám vah  $w_{ij}$  v čase  $k$  zvýšené o součin vstupního stavu neuronu  $x_j$ , výstupu z neuronu  $y$  a koeficientem rychlosti učení  $\alpha$  [DOLEŽEL, 2008].

### 2.3.2 Chybová učení

Chybové učení je učení s učitelem. Obecně platí následující vztahy

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad , \quad (2.21)$$

kde pak  $\Delta w_{ij}(k)$  bude

$$\Delta w_{ij}(k) = \alpha \cdot y_i(k) \cdot [y_{sj}(k) - y_{mj}(k)] \quad (2.22)$$

kde adaptace vah spojení  $\Delta w_{ij}$  v čase  $k$  mezi neuronem  $i$  a neuronem  $j$  je rovna součinu hodnoty signálu z neuronu  $i$  do neuronu  $j$  vstupu  $x_i$  a rozdílu požadované hodnoty

výstupního signálu z neuronu  $y_{sj}$  a skutečného výstupu  $y_{Mj}$  násobeného ještě koeficientem rychlosti učení  $\alpha$  [DOLEŽEL, 2008].

Takto nelze chybové učení aplikovat na vícevrstvé sítě, protože není možno zjistit hodnoty chyb na výstupu ze skrytých vrstev neuronové sítě. Proto byly navrženy modifikace, které jsou dále popsány v následujících kapitolách [DOLEŽEL, 2008].

### **Algoritmus střední kvadratické chyby**

LMS algoritmus (Least Mean Square Algorithm) slouží k učení neuronových sítí bez skrytých vrstev, přičemž neurony mají agregační funkci definovanou vztahem (2.2) a aktivační funkce je lineární. Snahou je minimalizovat kritériální funkci definovanou vztahem

$$E(k) = \frac{1}{2} e^2(k) \quad (2.23)$$

$$\text{kde } e(k) = y_s(k) - y_M(k) \quad (2.24)$$

$$y(k) = x(k)^T \cdot w^0(k) \quad (2.25)$$

kde  $E$  je kritériální funkce,  $e$  je chyba mezi výstupem regulované soustavy  $y$  a neuronovým modelem  $y_M$ , přičemž skutečná hodnota se za předpokladu, že vstup je roven výstupu zjednoduší a stanoví jako skalární součin vektorů vstupů do neuronu  $x(k)$  a vektoru vah  $w^0(k)$  [DOLEŽEL, 2008].

### **Algoritmus se zpětným šířením chyby**

BPG algoritmus (Backpropagation Gradient Descent Alorithm) byl navržen pro neuronové sítě, které mají nejméně jednu skrytou vrstvu a tudíž na rozdíl od LMS algoritmu řeší problém nemožnosti přímého zjištění chyby na výstupu neuronu ve skryté vrstvě. Jedná se o zobecněný LMS algoritmus.

Princip je takový, že na vstup neuronové sítě je přivedena matice vstupních hodnot, která se porovná z požadovanou hodnotou na výstupu, určí se funkce a provede se zpětný výpočet přes všechny vrstvy neuronové sítě zpět ke vstupu neuronové sítě a adaptují se váhy spojení. Nevýhodou tohoto algoritmu je velká citlivost na tréninková data a inicializace parametrů vah spojení neuronové sítě.

Chybu na výstupního  $j$ -tého neuronu lze určit dle vztahu

$$e_j(k) = y_{sj}(k) - y_{mj}(k) . \quad (2.26)$$

Je-li kritériální funkce neuronu  $j$ -tý neuron  $k$  definovaná vztahem (2.27)

$$\varepsilon_j(k) = \frac{1}{2} e_j^2(k) , \quad (2.27)$$

pak celková chybová energie všech neuronů ve výstupní vrstvě bude vyjádřena následujícím vztahem

$$E(k) = \frac{1}{2} \sum_j e_j^2(k) . \quad (2.28)$$

Za předpokladu existence  $N$  uspořádaných dvojic tréninkových dat [vstup, výstup], pak hodnota kritériální funkce v rámci jedné epochy učení bude vyjádřena vztahem (2.29) [DOLEŽEL, 2008].

$$E_{AV} = \frac{1}{N} \sum_{k=1}^N E(k) . \quad (2.29)$$

### **Kvazi-Newtonova metoda**

Kvazi-Newtonova metoda využívá Newtonův optimalizační algoritmus, jehož základní vztah pro adaptaci vah spojení je vyjádřen vztahem [DOLEŽEL, 2008]

$$\Delta w(k) = -H^{-1}(k) \cdot g(k) . \quad (2.30)$$

Zatímco BPG algoritmus využívá lineární aproximaci kritériální funkce v okolí pracovního bodu  $w(k)$ , Kvazi-Newtonova metoda použije aproximace vyššího řádu.

[DOLEŽEL, 2008]

Omezení této metody jsou:

- Náročnost výpočtu Hessovy matice  $H^{-1}$
- Určení Hessovy matice  $H^{-1}$  je možné za předpokladu, že matice  $H$  je regulární a tato podmínka zpravidla nebývá splněna.

V praxi se hlavně využívá modifikace (zjednodušení) Newtonovy metody tzv. Kvazi-Newtonova metoda, která řeší některé z těchto problémů. V matici  $H$  zanedbává

nediagonální prvky. Tím je zajištěno zvýšení rychlosti a stability výpočtu [DOLEŽEL, 2008].

### Levenbergův-Marquardtův algoritmus

Jde o kombinaci BPG algoritmu a Kvazi-Newtonovy metody. Tento algoritmus předpokládá účelovou funkci ve tvaru sumy čtverců, což je pro trénování dopředných vícevrstvých neuronových sítí typické. Za tohoto předpokladu může být Hessova matice  $H$  aproximována pomocí Jacobiovy matice, což vyplývá z následujícího vztahu

[DOLEŽEL, 2008].

$$H = J^T J \quad . \quad (2.31)$$

Algoritmu učení lze vyjádřit hodnotou změny vah spojení

$$\Delta w(k) = -[J^T(k)J(k) + \alpha I(k)]^{-1} g(k) \quad . \quad (2.32)$$

Jestliže se hodnota  $\alpha$  blíží nule, přechází Levenbergův-Marquardtův algoritmus na kvazi-Newtonovu metodu, ale pokud je  $\alpha$  velké, blíží se BPG algoritmu s nízkou hodnotou rychlosti učení. Parametr  $\alpha$  se adaptivně mění tak, že po každém úspěšném kroku, což znamená, že se kriteriální funkce sníží se hodnota parametru  $\alpha$  sníží a zvýší se za předpokladu, že byla kriteriální funkce v aktuální iteraci zvýšena [DOLEŽEL, 2008].

Tento algoritmus je nejrychlejší z výše uvedených algoritmů trénování. Je náročnější na paměť. Hodí se spíše na jednodušší topologie sítí [DOLEŽEL, 2008].

## **2.4 Inicializace váhových parametrů**

Každá procedura učení začíná inicializací váhových parametrů. Volba počátečních vah spojení zajistí, že síť bude dosahovat globální popř. lokální minimum kriteriální funkce během procesu učení. Pokud počáteční hodnoty vah definují počáteční bod tréninkového algoritmu, který ovlivňuje rychlost trénování, závisí na tom, zda tento počáteční bod míří ke globálnímu minimu nebo jen k lokálnímu.

### **2.4.1 Náhodná inicializace**

Před procesem učení se počáteční váhy spojení nastaví na hodnotu v polovině rozsahu hodnot  $(0+k)/2$ . Následkem tohoto řešení je, že se předložené vzory podobné počátečnímu nastavení náhodně rozdělují mezi ještě nenaučené [ŠKUTOVÁ, 2004].

Během trénování je třeba sledovat jeho průběh. To se děje pomocí chybové funkce, která může být vyjádřena např. pomocí závislosti součtu kvadratických chyb SSE (Sum Squared Error) resp. střední kvadratické chyby MSE (Mean Squared Error) na počtu iterací (učebních etap, epoch), které jsou dány

$$SSE = \sum_{i=1}^{n_{ep}} (y_s - y_M)^2 = \sum_{i=1}^{n_{ep}} e_i^2 \quad (2.33)$$

$$MSE = \frac{1}{N} \sum_{i=1}^{n_{ep}} (y_s - y_M)^2 = \sum_{i=1}^{n_{ep}} \frac{e_i^2}{N} \quad (2.34)$$

kde  $n_{ep}$  je počet trénovacích epoch,  $y_M$  výstup neuronové sítě a  $y_s$  je požadovaná hodnota. Váhy spojení se inicializují náhodně v malém rozsahu např.  $[-0,3; 0,3]$ .

#### 2.4.2 Nguyen – Windrow inicializace

Využívá se často pro inicializace dopředných neuronových sítí. Tato metoda provede vhodné rozdělení počátečních vah spojení u skrytých neuronů tak, že se učení stává efektivnější. Pro neuronovou dopřednou síť s  $n$  vstupy a  $p$  skrytými neurony se stanoví faktor  $\beta$ , který je vyjádřen následující rovnicí

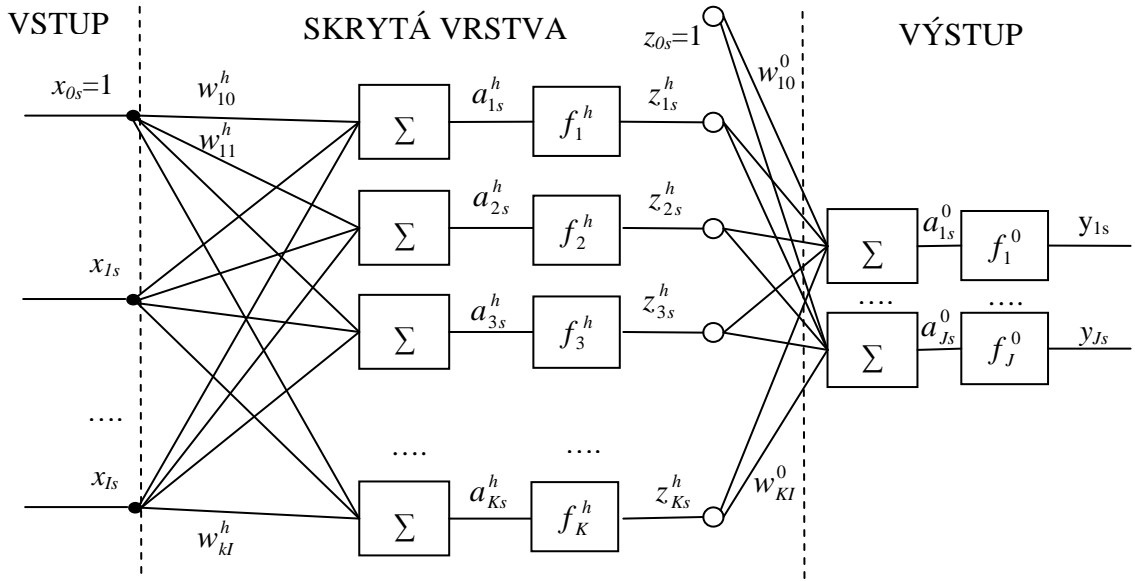
$$\beta = 0,7 \cdot \left( p^{\frac{1}{n}} \right) . \quad (2.35)$$

Následuje náhodné stanovení vah spojení  $w_i$  v intervalu  $[-1, 1]$ . Tyto náhodně vygenerované počáteční váhy se modifikují podle vztahu (2.36) a následně jsou použity jako počáteční hodnoty algoritmu učení. Pro výstupní vrstvu se generují hodnoty z intervalu  $[-0,5; 0,5]$  [ŠKUTOVÁ, 2004].

$$w_i^0 = \frac{\beta \cdot w_i^{\text{random}[-1,1]}}{\left\| w_i^{\text{random}[-1,1]} \right\|} \quad (2.36)$$

#### 2.4.3 Metoda inicializace založená na citlivostní analýze

Tato metoda využívá citlivosti všech parametrů každé vrstvy vzhledem k jejich vstupům a výstupům. [ŠKUTOVÁ, 2008]



Obr. 12 Dvouvrstvá dopředná neuronová síť

Síť zobrazená na obr. 12 se rozdělí na dvě jednovrstvé sítě. První síť má  $I$  vstupů  $x_{is}$  a  $K$  výstupů  $z_{ks}$  a druhá síť má  $K+1$  vstupů  $z_{ks}$  a  $J$  výstupů  $y_{js}$ ,  $f_k^h$  je aktivační funkce ve skryté vrstvě a  $f_j^o$  je aktivační funkce výstupní vrstvy neuronové sítě. Pokud je výstup z prostřední vrstvy  $z_{ks}$  známý, je kritériální funkce definována takto [ŠKUTOVÁ, 2008]

$$E(y) = E^h(y) + E^o(y) = \sum_{s=1}^S \left[ \sum_{k=1}^K \left( \sum_{i=0}^I w_{ki}^h x_{is} - f_k^h(z_{ks}) \right)^2 + \sum_{j=1}^J \left( \sum_{k=0}^K w_{jk}^0 z_{ks} - f_j^o(y_{js}) \right)^2 \right] \quad (2.37)$$

Na počátku algoritmu je nutné nastavit počáteční hodnoty  $z_{ks}$  náhodně a to z intervalu  $[0,05;0,95]$ . Dále je nutné sestavení soustavy lineárních rovnic, které budou modifikovat hodnoty vah neuronové sítě na základě účelové funkce. Jednotlivé citlivostní funkce jsou dány následujícím vztahem

$$\frac{\partial E}{\partial z_{ks}} = \frac{\partial E^h}{\partial z_{ks}} + \frac{\partial E^o}{\partial z_{ks}} = \frac{2 \left( \sum_{i=0}^I w_{ki}^h x_{is} - f_k^h(z_{ks}) \right)}{f_k^h(z_{ks})} + 2 \sum_{j=1}^J \left( \sum_{m=0}^K w_{jm}^0 z_{ms} - f_j^o(y_{js}) \right) w_{jk}^0, \quad (2.38)$$

kde  $k = 1, \dots, K$  a  $z_{ks} = 0$  pro všechna  $K$ . Při použití aproximace Taylorovou řadou bude kritériální funkce vyjádřena vztahem



$$E(z + \Delta z) = E(z) + \sum_{k=1}^K \sum_{s=1}^S \frac{\partial E}{\partial z_{ks}} \Delta z_{ks} \approx 0 \quad (2.39)$$

a po následné úpravě dostaneme hodnotu výstupu skryté vrstvy

$$\Delta z = -\rho \frac{E(z)}{\|\nabla E\|^2} \nabla E \quad (2.40)$$

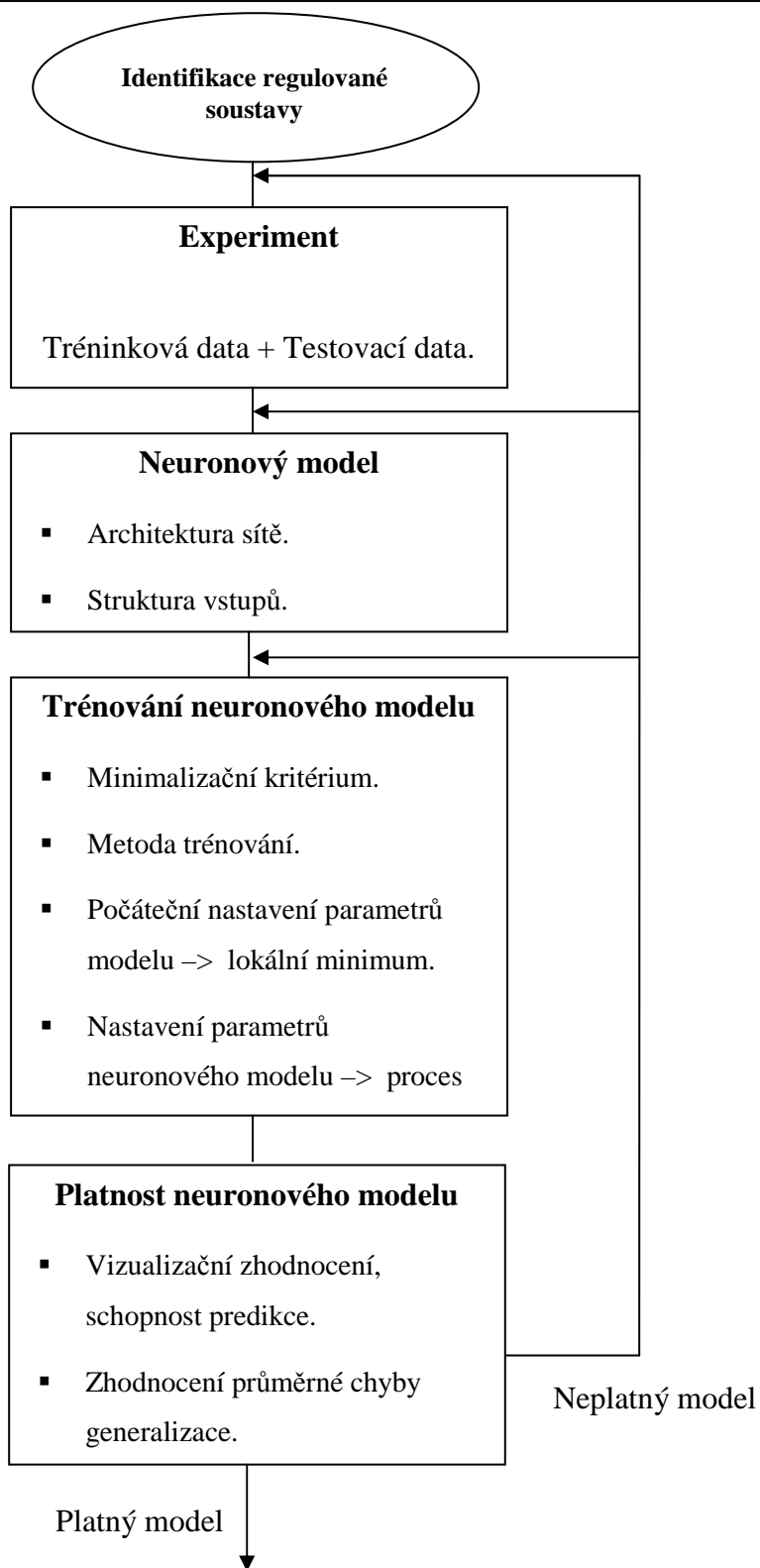
kde  $\rho$  je velikost kroku inicializace. Tento algoritmus inicializace vyžaduje několik cyklů výpočtu hodnot vah spojení a aktualizací výstupní hodnoty skryté vrstvy na základě kritériální funkce.

### **3 Identifikace regulované soustavy s využitím neuronových sítí**

Identifikace spočívá v odvození matematického popisu dynamického systému z měření na dané regulované soustavě. Pokud je identifikace založená výhradně na naměřených datech, hovoříme o modelování černé skříňky. Ovšem jakákoliv znalost systému usnadňuje jeho identifikaci. Za základní znalosti můžeme považovat:

- Řád regulované soustavy.
- Dynamika regulované soustavy (rychlá, pomalá).
- Stabilita.
- Vzorkovací frekvence.
- Pracovní rozsah.
- Dopravní zpoždění.

Identifikace neuronovou sítí se provádí podle předem stanoveného postupu viz obr. 13. Je sestavena do čtyř fází: experiment, volba struktury neuronového modelu, proces trénování a v poslední fázi se jedná o zhodnocení platnosti neuronového modelu v maximální shodě s modelovaným systémem.



Obr. 13 Procedura identifikace systému s využitím neuronové sítě [ŠKUTOVÁ, 2004]

V následujících podkapitolách je uveden popis jednotlivých kroků identifikační procedury pro získání platného modelu zastoupeného vícevrstvou perceptronovou sítí obr. 13.

### 3.1 Tréninková množina dat

Tréninková množina dat je složena z vektorů požadovaných vstupních a výstupních hodnot, které umožní nastavit parametry neuronové sítě tak, aby měla požadované vlastnosti. Rozsah vstupního signálu a následně získaného výstupního signálu by měl obsahovat hodnoty z celé pracovní oblasti. Pak bude zajištěna platnost modelu v daném rozsahu hodnot. Velikost tréninkové množiny ovlivňuje kvalitu trénování neuronové sítě.

[ŠKUTOVÁ, 2004]

Nyní si uvedeme typy vstupních signálů, který reprezentuje vlastnosti a chování neuronové sítě a tudíž i vlastnosti dané identifikované soustavy [ŠKUTOVÁ, 2004].

Vstupní signály tvoří:

- Přímka.
- Sinusový signál.
- Obdélníkový signál.
- Pilový signál.
- Náhodný signál.

popř. jejich kombinace.

### 3.2 Struktura modelu

V této podkapitole je uveden rozbor struktury modelů určené pro identifikaci nelineárních dynamických systémů. V případě volby struktury se postupuje ve dvou krocích:

1. Volba skupiny struktury modelu, který odpovídá identifikovanému systému.  
Např. RBF sítě, lineární model atd.
2. Výběr podskupiny struktury modelů, např. ARX, NARX atd.

[ŠKUTOVÁ, 2004]

#### Struktury lineárních modelů

Systém je popsán následující rovnicí

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k) \quad (3.1)$$

kde  $\mathbf{G}$  a  $\mathbf{H}$  jsou matice v závislosti na operátoru zpětného posunutí  $q^{-1}$ . Proměnná  $e(k)$  je signál bílého šumu, který je nezávislý na předchozích vstupních hodnotách. V případě, že se jedná o mnohorozměrný systém jsou proměnné  $\mathbf{u}(k)$ ,  $\mathbf{y}(k)$  a  $e(k)$  vektory a  $\mathbf{G}$  a  $\mathbf{H}$  matice [ŠKUTOVÁ, 2004].

Obecný tvar struktury modelu je dán vztahem

$$A(q^{-1})y(k) = q^{-d} \frac{B(q^{-1})}{F(q^{-1})} u(k) + \frac{C(q^{-1})}{D(q^{-1})} e(k) \quad , \quad (3.2)$$

kde polynomy jsou ve tvaru

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_m q^{-m} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_k q^{-k} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_l q^{-l} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \dots + f_r q^{-r} \end{aligned}$$

### Lineární modely

#### *FIR struktura modelu (The Finite Impulse Response Model Structure)*

Jedná se o nejjednodušší typ struktury modelu a její popis je dán vztahem

$$A(q^{-1})y(k) = q^{-d} B(q^{-1})u(k) + e(k) \quad (3.3)$$

a predikce je dána rovnicí

$$\hat{y}(k | \theta) = q^{-d} B(q^{-1})u(k), \quad (3.4)$$

což může být vyjádřeno v regresním tvaru takto

$$\hat{y}(k | \theta) = \varphi^T(k) \theta, \quad (3.5)$$

a regresní vektor  $\varphi(k)$  je definován následujícím vztahem

$$\varphi(k) = [u(k-d), \dots, u(k-d-m)]^T, \quad (3.6)$$

kde  $d$  je dopravní zpoždění.

**ARX struktura modelu (Autoregressive with eXternal input)**

Jedná se o autoregresivní model s externím vstupem a tuto strukturu lze definovat takto

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + e(k) \quad (3.7)$$

a predikci výstupu lze definovat takto

$$\hat{y}(k | \theta) = q^{-d}B(q^{-1})u(k) + [1 - A(q^{-1})]y(k) = \varphi^T(k)\theta \quad (3.8)$$

kde regresní vektor  $\varphi(k)$  je dán následujícím vztahem

$$\varphi(k) = [y(k-1), \dots, y(k-n), u(k-d), \dots, u(k-d-m)]^T \quad (3.9)$$

**ARMAX struktura modelu (AutoRegressive Moving Average with eXternal Input)**

Jde o model nazývaný autoregresivní model s klouzavým průměrem a s dalším vstupem, a jeho strukturu lze popsat vztahem

$$A(q^{-1})y(k) = q^{-d}B(q^{-1}) \cdot u(k) + C(q^{-1}) \cdot e(k), \quad (3.10)$$

predikce výstupu má tvar

$$\begin{aligned} \hat{y}(k | \theta) &= q^{-d} \frac{B(q^{-1})}{C(q^{-1})} u(k) + \left[ 1 - \frac{A(q^{-1})}{C(q^{-1})} \right] y(k) \\ &= q^{-d} B(q^{-1}) u(k) + [1 - A(q^{-1})] y(k) + [C(q^{-1}) - 1] \varepsilon(k, \theta), \\ &= \varphi^T(k, \theta) \theta \end{aligned} \quad (3.11)$$

kde

$$\varepsilon(k | \theta) = y - \hat{y}(k | \theta) \quad (3.12)$$

představuje chybu predikce.

Regresní vektor  $\varphi(k)$  bude popsán takto

$$\varphi(k | \theta) = [y(k-1), \dots, y(k-n), u(k-d), \dots, u(k-d-m), \varepsilon(k | \theta), \dots, \varepsilon(k-K | \theta)]^T \quad (3.13)$$

**OE struktura model (Output Error Model Structure)**

Využití tohoto modelu je jen tehdy, pokud bílý šum lze nahradit konkrétním šumem měření. Struktura modelu je popsána následujícím vztahem

$$y(k) = q^{-d} \frac{B(q^{-1})}{F(q^{-1})} u(k) + e(k) \quad (3.14)$$

predikce výstupu pro tento systém má tvar

$$\begin{aligned} y(k | \theta) &= q^{-d} \frac{B(q^{-1})}{F(q^{-1})} u(k) \\ &= q^{-d} B(q^{-1}) u(k) + [1 - F(q^{-1})] \hat{y}(t | \theta) \\ &= \varphi^T(k, \theta) \theta \end{aligned} \quad (3.15)$$

kde  $\theta$  je vektor parametrů a pak regresní vektor  $\varphi(k)$  bude popsán takto

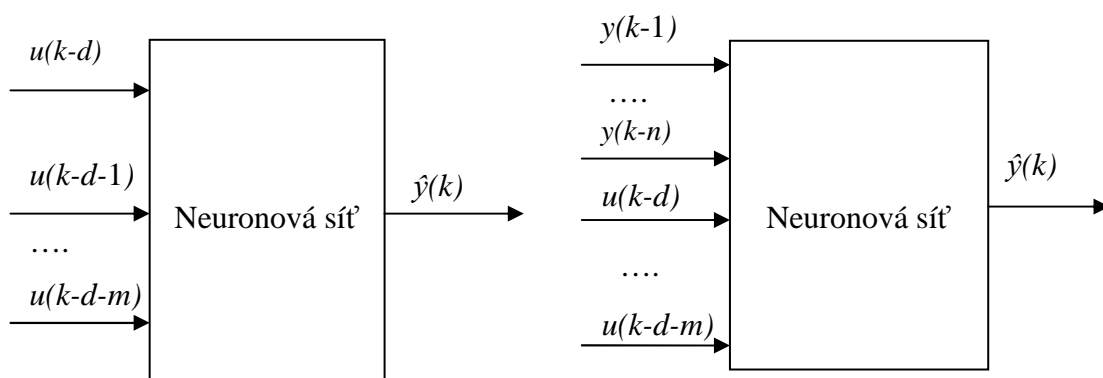
$$\varphi(k | \theta) = [\hat{y}(t-1 | \theta), \dots, \hat{y}(t-d | \theta), u(t-d), \dots, u(t-d-m)]^T \quad (3.16)$$

**Struktury nelineárních modelů**

Vícevrstvé perceptronové sítě jsou svojí vnitřní architekturou a nelineárními aktivačními funkcemi vhodné pro učení nelineárních systémů. Pak je možné využít obdobných regresních vektorů lineárních modelů. Tyto modely jsou pak díky nelineárnímu typu modelu neuronové sítě označeny jako nelineární. V další části jsou uvedeny nelineární modely NFIR, NARX, NARMAX a NOE [ŠKUTOVÁ, 2004].

**Nelineární modely****NFIR a NARX struktura modelu**

U struktury NFIR (Nonlinear Finite Impulse Response) jsou prediktory vždy stabilní, protože se jedná o algebraický vztah mezi poslední vstupní a výstupní hodnotou. Struktura NFIR je na obr. 15 a struktura NARX (Nonlinear Autoregressive with eXternal Input) na obr. 14.



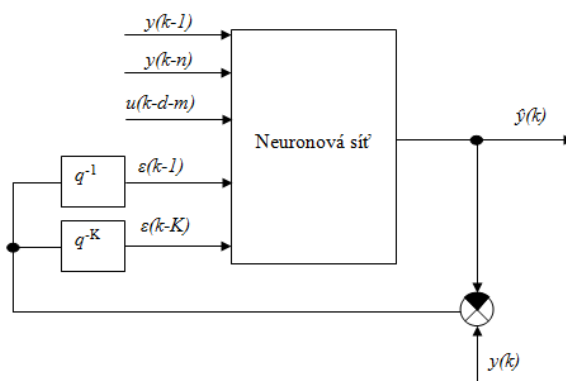
Obr. 15 NFIR struktura modelu [ŠKUTOVÁ, 2004]

Obr. 14 NARX struktura modelu [ŠKUTOVÁ, 2004]

### NARMAX struktura modelu

Strukturu NARMAX (Nonlinear AutoRegressive Moving Average with eXternal Input) můžeme popsat následujícím vztahem

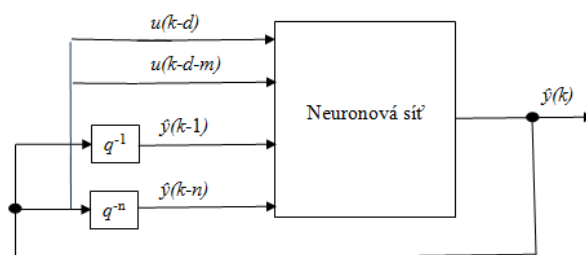
$$\varphi(k, \theta) = [y(k-1), \dots, y(k-n), u(k-d), \dots, u(k-d-m), \varepsilon(k-1), \varepsilon(k-K)]^T \quad (3.17)$$



Obr. 16 NARMAX struktura modelu [ŠKUTOVÁ, 2004]

### NOE struktura modelu

Struktura NOE (Nonlinear Output Error Model Structure) je na obr. 17.



Obr. 17 NOE struktura modelu [ŠKUTOVÁ, 2004]

Prediktory musí být stabilní. Nestabilita může nastat u systémů se zpětnou vazbou. Proto je vhodnější zvolit struktury NFIR a NARX, které jsou stabilní [ŠKUTOVÁ, 2004].



## 4 Identifikace regulovaných soustav

Identifikace regulovaných soustav s využitím neuronových sítí směřuje k tvorbě dopředných nebo inverzních modelů, které jsou dále aplikovány v metodách řízení. Pokud je cílem najít požadovaný inverzní či dopředný model, je nutné vhodným způsobem zvolit strukturu neuronové sítě a další uvažované parametry identifikace.

Kapitola má za cíl sledovat, jaké volitelné parametry odpovídají vybraným typům regulovaných soustav. Pokud je identifikace modelů úspěšná, pak je možné je dále aplikovat do oblasti řízení. V této kapitole je provedena podrobná analýza a výsledky identifikace jsou prokázány na dvou vybraných typech regulovaných soustav.

### 4.1 Proporcionální soustava se setrvačností 3. řádu

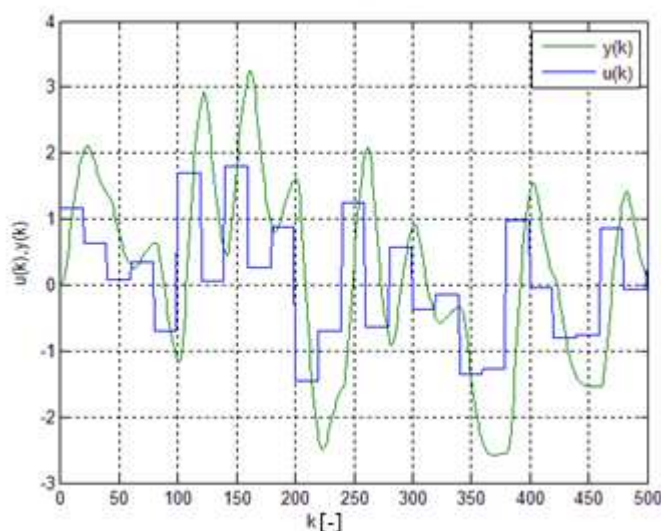
Soustava je popsána následujícím diferenční rovnicí

$$y(k) - 2,3364 \cdot y(k-1) + 1,8196 \cdot y(k-2) - 0,4724 \cdot y(k-3) = 0,0043 \cdot u(k-1) + 0,0144 \cdot u(k-2) + 0,003 \cdot u(k-3). \quad (4.1)$$

Vzorkovací perioda byla zvolena  $T = 1$  s a čas simulace  $t = 500$  s. Tak získáme 500 vzorků pro trénování modelu neuronové sítě. Následující tabulka tab. 1 zobrazuje vlastnosti identifikace pro systém popsany diferenční rovnicí.

Tab. 1 Vlastnosti identifikace pro proporcionální soustavu se setrvačností 3. řádu popsany diferenční rovnicí

Vlastnosti identifikace pro daný systém	
Architektura sítě	Dvouvrstvá perceptronová síť
Počet neuronů ve skryté vrstvě	10
Aktivační funkce	Hyperbolicko-tangenciální aktivační funkce, Lineární aktivační funkce
Struktura modelu	NARX
Tréninkový algoritmus	Levenbergův-Marquardtův algoritmus
Počet natrénovaných modelů	30
Počet tréninkových epoch	5000
Budící signál tréninkové množiny dat	Náhodný signál se změnou úrovně $u(k)$ po 20 s
Počet vstupů $m$	1 až 3
Počet výstupů $n$	3 až 6



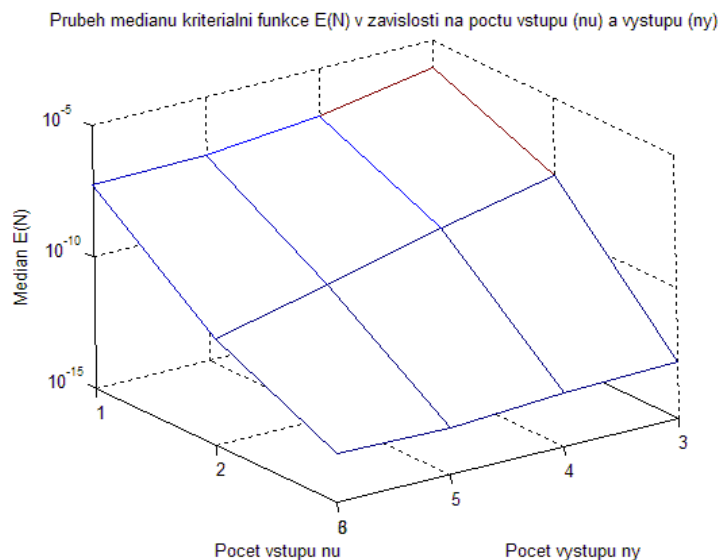
Obr. 18 Tréninková množina dat

Následující tabulka ukazuje medián hodnot kritériální funkce  $E(5000)$  v závislosti na kombinaci počtu vstupů  $m$  a výstupů  $n$  s příslušnou predikcí. Kritériální funkce určuje hodnotu ve fázi trénování inverzního modelu po 5000 hodnotách.

Tab. 2 Medián kritériální funkce  $E(5000)$  pro dopředné modelování proporcionální soustavu 3. řádu

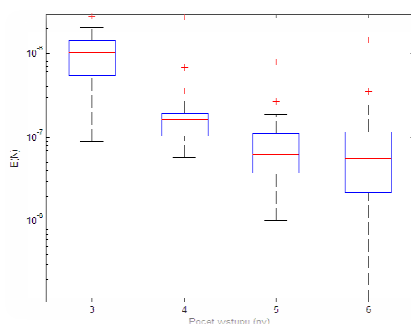
Medián hodnot kritériální funkce $E(5000)$				
Proporcionální soustava se setrvačností 3. řádu				
$m \backslash n$	3	4	5	6
1	$1,03 \cdot 10^{-6}$	$1,62 \cdot 10^{-7}$	$6,20 \cdot 10^{-8}$	$5,71 \cdot 10^{-8}$
2	$1,16 \cdot 10^{-8}$	$1,30 \cdot 10^{-9}$	$1,14 \cdot 10^{-10}$	$1,13 \cdot 10^{-11}$
3	$1,32 \cdot 10^{-13}$	$1,04 \cdot 10^{-13}$	<b><math>6,18 \cdot 10^{-14}</math></b>	$7,65 \cdot 10^{-14}$

Z tab. 2 je vidět, že pro modelování vybrané regulované soustavy je nejlepší volbou počtu minulých hodnot vstupů a výstupů 3 vstupy a 5 výstupů. Je vidět, že tendence chyby je klesající pro zvyšující se počet vstupů do neuronové sítě (obr. 19).

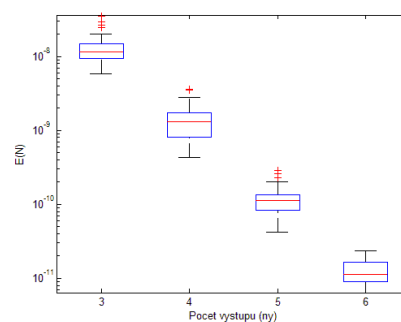


Obr. 19 Průběh mediánu kritériální funkce  $\tilde{E}(5000)$  trénování dopředného modelu pro proporcionální soustavu 3. řádu

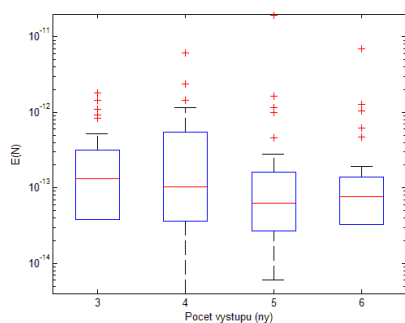
Pro zhodnocení nastavení počtu minulých hodnot vstupních a výstupních hodnot identifikovaného modelu bylo pro každou kombinaci vždy trénováno 30 kandidátů na požadovaný model a v grafech jsou pak zobrazeny mediány kritériálních funkcí získaných na konci procesu trénování z důvodu odstranění vlivu počáteční inicializace neuronové sítě.



Obr. 20 Statistické zhodnocení kritériální funkce  $E(5000)$  pro  $m=1$  a různý počet výstupních hodnot  $n$

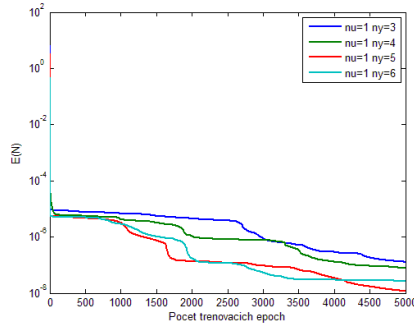


Obr. 21 Statistické zhodnocení kritériální funkce  $E(5000)$  pro  $m=2$  a různý počet výstupních hodnot  $n$

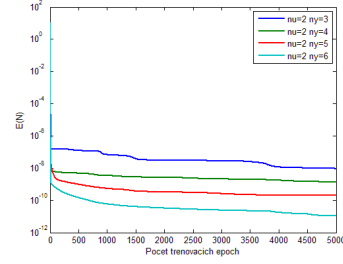


Obr. 22 Statistické zhodnocení kritériální funkce  $E(5000)$  pro  $m=3$  a různý počet výstupních hodnot  $n$

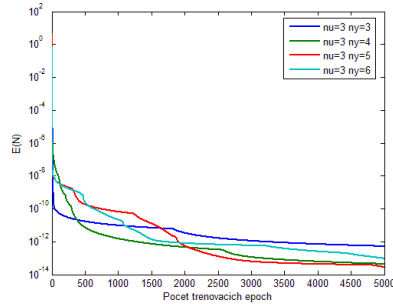
Porovnání průběhů kritériální funkce trénování dopředného modelu s její nejnižší hodnotou ze všech 30 trénovaných modelů je zobrazeno na obr. 23 až obr. 25.



Obr. 23 Průběh kritériální funkce  $E(N)$  při trénování (učení) pro modely s počtem minulých vstupních signálů  $m = 1$  a různým počtem minulých výstupních veličin



Obr. 24 Průběh kritériální funkce  $E(N)$  při trénování (učení) pro modely s počtem minulých vstupních signálů  $m = 2$  a různým počtem minulých výstupních veličin



Obr. 25 Průběh kritériální funkce  $E(N)$  při trénování (učení) pro modely s počtem minulých vstupních signálů  $m = 3$  a různým počtem minulých výstupních veličin

Na základě předchozího testování počtu vstupů  $n_u$  a počtu výstupů  $n_y$  byl vybrán model, kde  $n = 5$  a  $m = 3$ .

Další analýza nastavení parametrů neuronové sítě se týká volby počtu skrytých neuronů za předpokladu použití struktury modelu NARX je regresním vektorem dán vztahem s tímto konkrétním nastavením

$$\varphi(k) = [y(k-1), \dots, y(k-5), u(k-1), \dots, u(k-3)]^T, \quad (4.2)$$

kde počet neuronů ve skryté vrstvě by měl být dán vztahem

$$p \geq m + n \quad (4.3)$$

kde  $m$  je počet vstupů a  $n$  je počet výstupů [DRÁBEK, SEIDL, TAUFER, 2006].

Pak nezbytný minimální počet skrytých neuronů je dán např. pro  $m = 3$  a  $n = 5$  vztahem

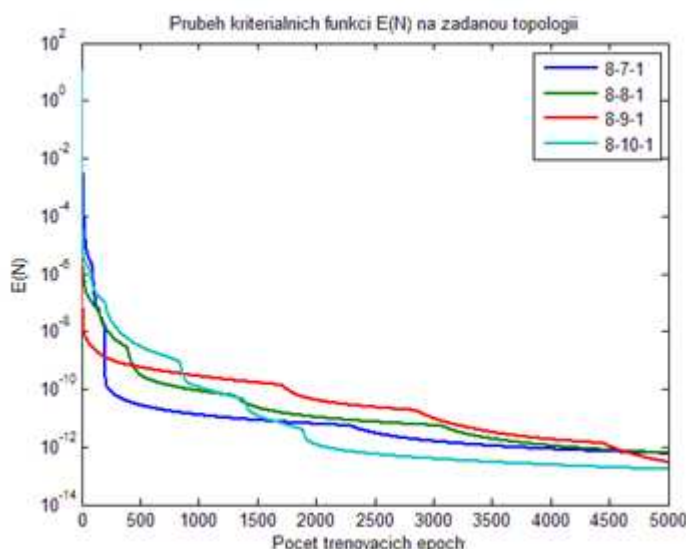
$$p \geq 8 + 1 \geq 9 \quad . \quad (4.4)$$

Bylo zvoleno  $p \in \langle 7 \div 10 \rangle$ .

Nejlepší hodnota kritériální funkce  $E(N)$  z 10-ti modelů je v následující tabulce. Průběhy nejlepších modelů každé architektury z tab. 3 jsou na obr. 26.

Tab. 3 Hodnota kritériální funkce  $E(5000)$  pro daný počet neuronů ve skryté vrstvě

Počet skrytých neuronů	Hodnota kritériální funkce $E(5000)$
7	$5,87 \cdot 10^{-13}$
8	$6,13 \cdot 10^{-13}$
9	$2,97 \cdot 10^{-13}$
10	$1,66 \cdot 10^{-13}$

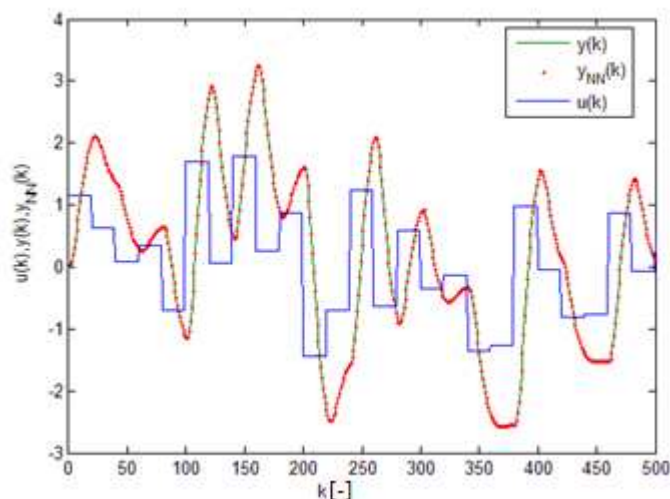


Obr. 26 Průběh trénování (učení) dopředného modelu regulované soustavy pro různé počty skrytých neuronů

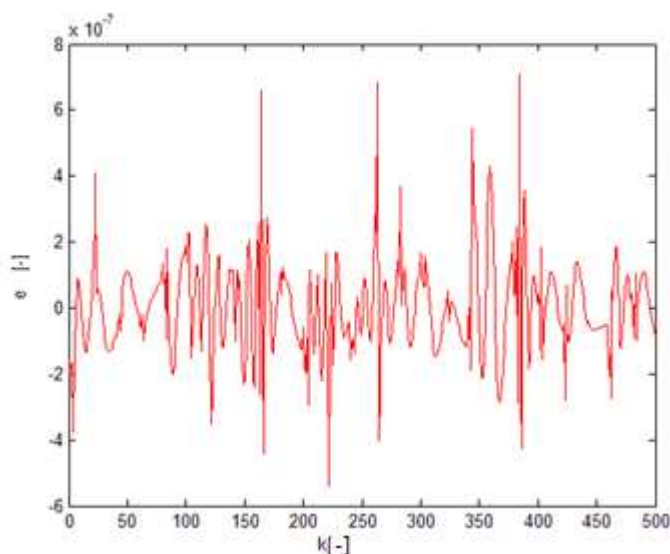
Na základě nejnižší hodnoty kritériální funkce (tab. 3) a rychlosti učení (obr. 26) byla zvolena struktura neuronové sítě 8–10–1. Takové nastavení neuronové sítě je dále využito ve fázi trénování dopředného modelu regulované soustavy. Následně bylo provedeno ověření platnosti dopředného modelu na testovacích datech. Kritériem platnosti modelu je hodnota chyby mezi výstupem regulované soustavy a výstupem dopředného modelu.

### Testování neuronové sítě s architekturou 8-10-1

Odezva dopředného modelu na výstupní data tréninkové množiny dat se téměř shoduje s požadovaným výstupem (obr. 27). Chyba výstupních veličin odpovídá hodnotě kritériální funkce, která byla minimalizována procesem trénování. Pro testování je nutné použít jinou množinu dat.



Obr. 27 Predikovaný výstup modelu neuronové sítě regulované soustavy na tréninkový signál



Obr. 28 Průběh chyby dopředného modelu testovaného množinou tréninkových dat

Největší hodnota chyby mezi výstupem regulované soustavy a výstupem dopředného modelu (obr. 28) je pro danou tréninkovou množinu dat dána hodnotou

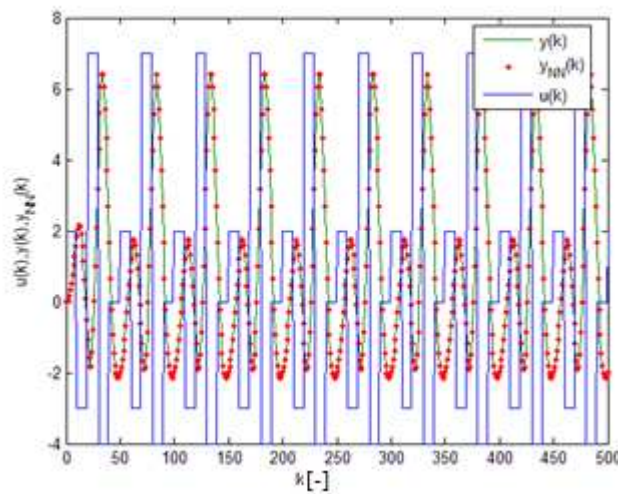
$$|e_{max}| = 7,07 \cdot 10^{-7} . \quad (4.5)$$

Mnohem důležitější jsou výsledky testování dopředného modelu na jiných datech než na tréninkových datech. Testovací signál má průběh stupňovité funkce, která má

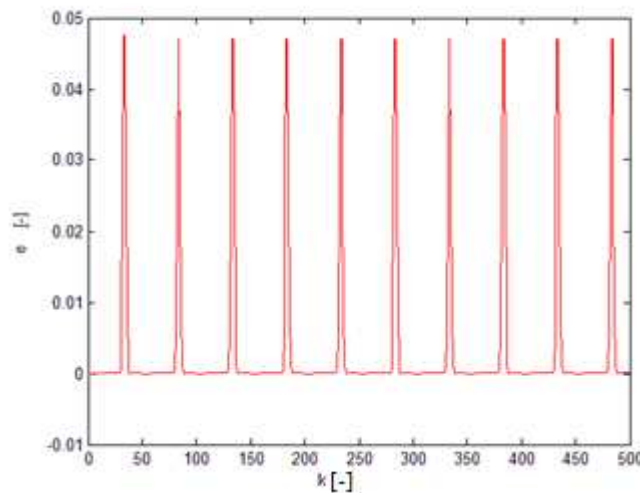
vzhledem k tréninkovému signálu složitější průběh tzn., že průběh má rychlejší změny akční veličiny  $u(k)$ . Stupňovitá funkce má tvar

$$u_{testovací}(k) = \begin{cases} 2 & 0 \leq k \leq 10 \\ -3 & 11 \leq k \leq 20 \\ 7 & 21 \leq k \leq 30 \\ -4 & 31 \leq k \leq 40 \\ 0 & 41 \leq k \leq 50 \end{cases} \quad (4.6)$$

a tento signál se cyklicky opakuje po dobu 500 vzorků simulace.



Obr. 29 Predikovaný výstup modelu neuronové sítě regulované soustavy na testovací signál



Obr. 30 Průběh chyby dopředného modelu testovaného množinou testovacích dat

Největší hodnota chyby mezi výstupem regulované soustavy a výstupem dopředného modelu (Obr. 30) je pro danou tréninkovou množinu dat dána hodnotou

$$|e_{max}| = 0,0477 \quad (4.7)$$

Jak se dalo předpokládat, tak maximální odchylka  $e_{max}$  mezi výstupem regulované soustavy a výstupem neuronové sítě pro tréninkovou množinu dat je menší než pro testovací množinu dat, protože testovací signál zasahuje i do hodnot, na něž nebyla síť natrénována.

## 4.2 Nelineární regulovaná soustava

Pro analýzu volitelnosti struktury neuronové sítě pro konkrétní typ regulované soustavy byla zvolena nelineární regulovaná soustava popsána následující diferenční rovnicí

$$y(k+1) - 0,6 \cdot y(k) = 0,2 \cdot u^2(k) \quad . \quad (4.8)$$

Jedná se o nelineární soustavu 1. řádu, jehož statická charakteristika má kvadratický průběh. Vzorkovací perioda  $T = 0,1$  s a čas simulace  $t = 80$  s. Tím získáme 800 vzorků pro stanovení tréninkové množiny dat neuronové sítě.

Tab. 4 Vlastnosti identifikace pro nelineární regulovanou soustavu popsanou diferenční rovnicí

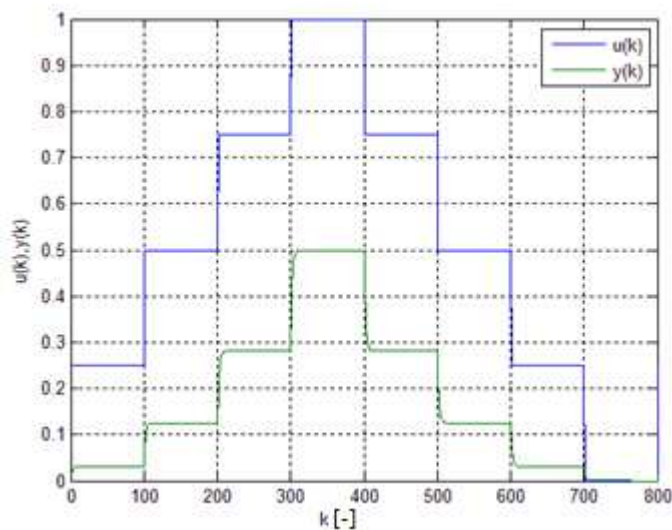
Vlastnosti identifikace pro systém popsaný diferenční rovnicí	
Architektura sítě	Dvouvrstvá perceptronová síť
Počet neuronů ve skryté vrstvě	5
Aktivační funkce	Hyperbolicko-tangenciální aktivační funkce, Lineární aktivační funkce
Struktura modelu	NARX
Tréninkový algoritmus	Levenbergův-Marquardtův algoritmus
Počet natrénovaných modelů	30
Počet tréninkových epoch	1000
Budící signál tréninkové množiny dat	Náhodný signál se změnou úrovně $u(k)$ po 10 s
Počet vstupů $m$	1 až 3
Počet výstupů $n$	3 až 6



Jako budící signál pro získání tréninkové množiny dat byla zvolena stupňovitá funkce ve tvaru

$$u_{\text{trénovací}}(k) = \begin{cases} 0,25 & 0 \leq k \leq 10 \\ 0,5 & 10,1 \leq k \leq 20 \\ 0,5 & 20,1 \leq k \leq 30 \\ 1 & 30,1 \leq k \leq 40 \\ 0,75 & 40,1 \leq k \leq 50 \\ 0,5 & 50,1 \leq k \leq 60 \\ 0,25 & 60,1 \leq k \leq 70 \\ 0 & 70,1 \leq k \leq 80 \end{cases} . \quad (4.9)$$

Na (obr. 31) je zobrazen průběh signálu tréninkové množiny dat nelineární regulované soustavy.



Obr. 31 Tréninková množina dat pro identifikaci dopředného modelu nelineární regulované soustavy

Tabulka (tab. 5) ukazuje medián hodnot kritériální funkce  $E(1000)$  v závislosti na kombinaci počtu vstupů  $m$  a výstupů  $n$  s příslušnou predikcí.

Tab. 5 Medián kriteriální funkce  $E(1000)$  pro nelineární regulovanou soustavu se setrvačností 1. řádu

Medián hodnot kriteriální funkce $E(1000)$				
Nelineární soustavu se setrvačností 1. řádu				
$m \backslash n$	1	2	3	4
1	$3,3068 \cdot 10^{-5}$	$3,3067 \cdot 10^{-5}$	$3,3066 \cdot 10^{-5}$	$3,3064 \cdot 10^{-5}$
2		$3,3067 \cdot 10^{-5}$	$3,3066 \cdot 10^{-5}$	$3,3065 \cdot 10^{-5}$
3			$3,3066 \cdot 10^{-5}$	$3,3064 \cdot 10^{-5}$

Z tab. 5 je vidět, že pro vybranou regulovanou soustavu by byla nejvhodnější kombinace 4 vstupů a 1 výstupu, protože má nejnižší hodnotu mediánu kriteriální funkce, ale protože se snažíme počet parametrů, které pro neuronovou síť nastavujeme, snižovat, byla zvolena struktura s jednou minulou vstupní a jednou minulou výstupní veličinou.

Další analýza struktury neuronové sítě z hlediska vlivu počtu skrytých neuronů na kvalitu identifikace je proveden pro model typu NARX s regresním vektorem

$$\varphi(k) = [y(k-1), u(k-1)]^T. \quad (4.10)$$

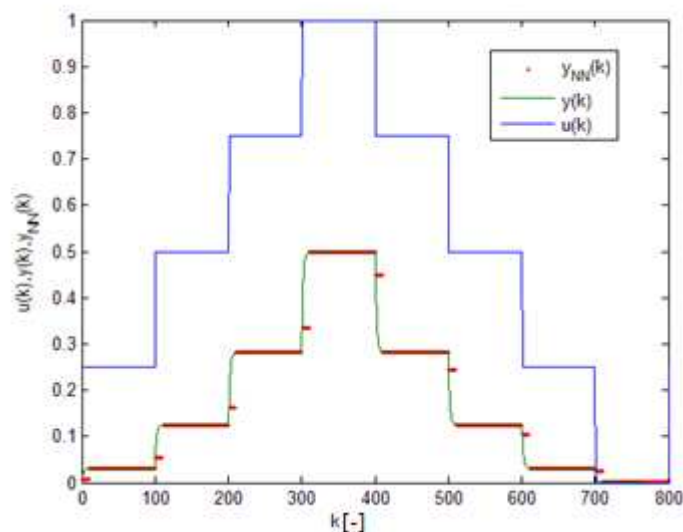
Pro stanovení počtu neuronů ve skryté vrstvě opět použijeme vzorec (4.3) a hodnoty kriteriálních funkcí pro 30 kandidátů na požadovaný model je v tab. 6.

Tab. 6 Tabulka závislosti hodnoty kriteriální funkce trénování na počtu skrytých neuronů

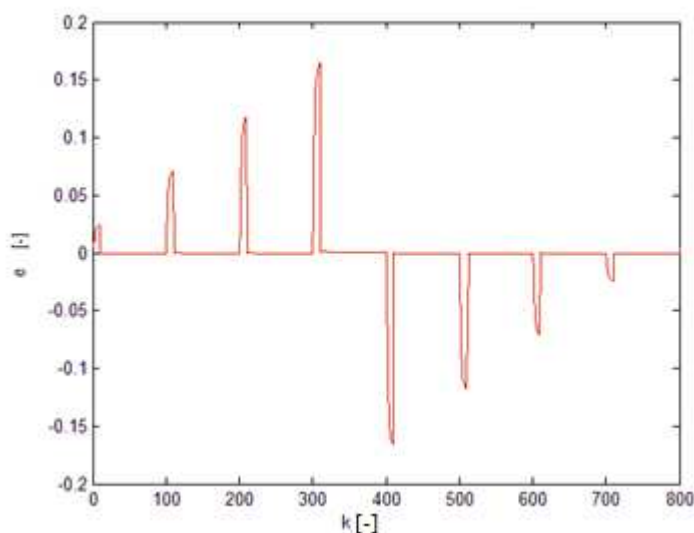
Počet neuronů ve skryté vrstvě	Hodnota kriteriální funkce $E(1000)$
3	$3,308 \cdot 10^{-5}$
4	$3,307 \cdot 10^{-5}$
5	$3,307 \cdot 10^{-5}$
6	$3,307 \cdot 10^{-5}$
7	$3,307 \cdot 10^{-5}$

Z tab. 6 vyplývá, že nejnižší hodnota kriteriální funkce  $E(1000)$  je dosázena při použití architektury 2–4–1. Při dalším zvyšování počtu neuronů ve skryté vrstvě se hodnota kriteriální funkce nemění. Nyní otestujeme neuronovou síť s architekturou 2–4–1 na tréninkovou a testovací množinu dat.

## Testování neuronové sítě se strukturou 2 – 4 – 1



Obr. 32 Predikovaný výstup modelu neuronové sítě regulované nelineární soustavy na tréninkový signál



Obr. 33 Průběh odchylky identifikace modelu regulované soustavy s využitím tréninkové množiny dat

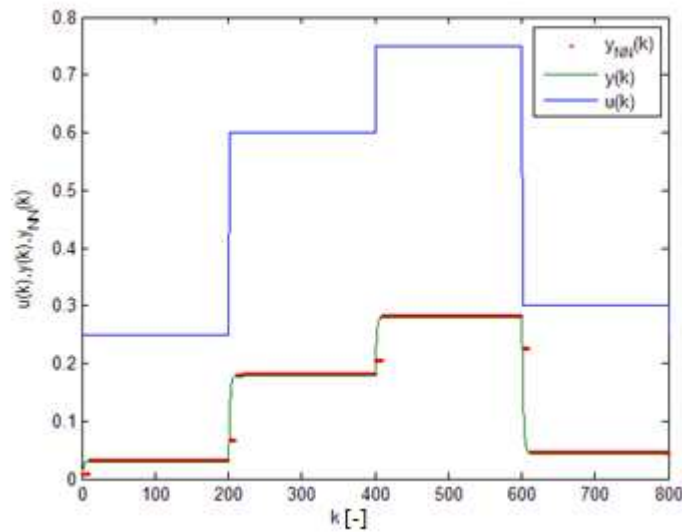
Z obr. 33 určíme největší hodnotu chyby mezi výstupem z regulované soustavy a výstupem z neuronové sítě tzn.  $|e_{max}|$  bude

$$|e_{max}| = 0,1653 . \quad (4.11)$$

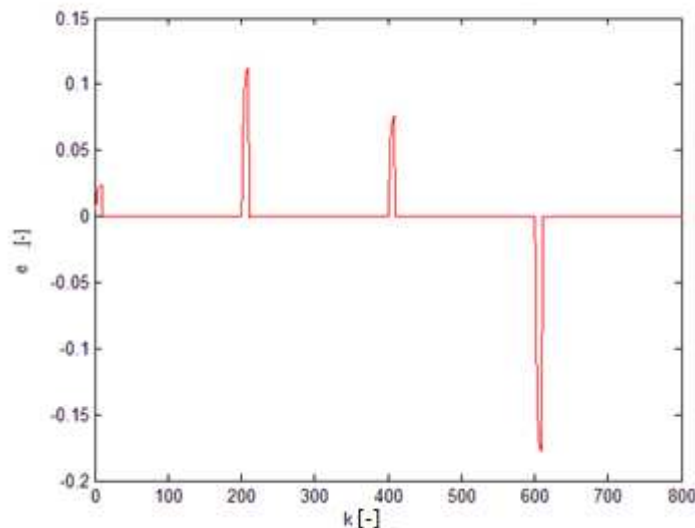
Ve fázi testování identifikovaného modelu byla aplikována testovací množina dat jejichž vstupní signál je ve tvaru stupňovité funkce dle vztahu

$$u_{testovací}(k) = \begin{cases} 0,25 & 0 \leq k \leq 200 \\ 0,6 & 200,1 \leq k \leq 400 \\ 1 & 400,1 \leq k \leq 600 \\ 0,75 & 600,1 \leq k \leq 800 \end{cases} . \quad (4.12)$$

Porovnáním průběhů regulované soustavy a identifikovaného dopředného modelu s využitím těchto testovacích dat je zřejmé z obr. 34 a vzhledem k velké shodě modelu s regulovanou soustavou je průběh odchylky této identifikace uveden na obr. 35.



Obr. 34 Odezva neuronové sítě na testovací signál pro nelineární regulované soustavu



Obr. 35 Průběh odchylky identifikace nelineární regulované soustavy na testovací množinu dat

Největší hodnota chyby mezi výstupem regulované soustavy a výstupem dopředného modelu (obr. 35) je pro danou tréninkovou množinu dat dána hodnotou

$$|e_{max}| = 0,1782 . \quad (4.13)$$

Simulace prokázaly, že stanovení parametrů neuronové sítě tzn. topologie, struktury modelu, počet vstupů (výstupů) atd., je velmi časově náročný proces. Proto tato práce měla směřovat k tomu, aby analyzovala proces návrhu struktury neuronové sítě alespoň pro časté typy regulovaných soustav, např. pro regulovanou soustavu typu proporcionální soustavy se setrvačností 3. řádu pro kterou je vhodné zvolit architekturu 8-10-1. Pro nelineární regulovanou soustavu se setrvačností 1. řádu je vhodné použít architekturu 2-4-1. Procesy trénování a testování byly provedeny v programu Matlab s podporou funkcí pro aplikaci neuronových sítí a pro ověření platnosti identifikovaného modelu byly sestaveny simulační schémata v programu Simulink, které jsou uvedeny v přílohách C a D.

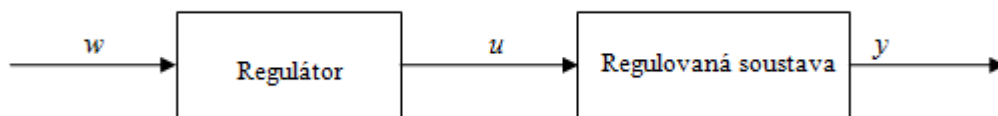
## 5 Řízení regulované soustavy s využitím neuronových sítí

Neuronové sítě se nejčastěji používají k řízení nelineárních systémů. U nelineárních systémů neplatí některé zákonitosti jako u lineárních soustav. Jde především o zákon superpozice tzn., že nedokáže předvídat budoucí dopad současných akčních zásahů. Vzhledem k různorodosti nelineárních systémů je nalezení obecného řešení téměř nemožné. Prvním krokem pro návrh neuronových regulátorů (inverzních modelů) je identifikace dané regulované soustavy pomocí neuronové sítě (dopředný model), z důvodu optimálního nastavení struktury neuronové sítě, jejího počtu minulých vstupních a výstupních veličin pro identifikaci regulované soustavy. Teprve pak je vhodné přistoupit k identifikaci inverzního modelu, který je aplikován ve strukturách metod řízení.

### 5.1 Přímé inverzní řízení pomocí neuronové sítě

Přímé inverzní řízení využívá vlastnosti inverzního modelu z předpokladu splnění vztahu

$$G_S \cdot G_{IM} = 1 \quad (5.1)$$



Obr. 36 Regulační obvod Přímého inverzního řízení

Pokud je soustava popsána diferenční rovnicí danou vztahem

$$y(k) = \varphi[y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)] \quad (5.2)$$

pak její inverzní model bude popsán diferenční rovnicí (5.3)

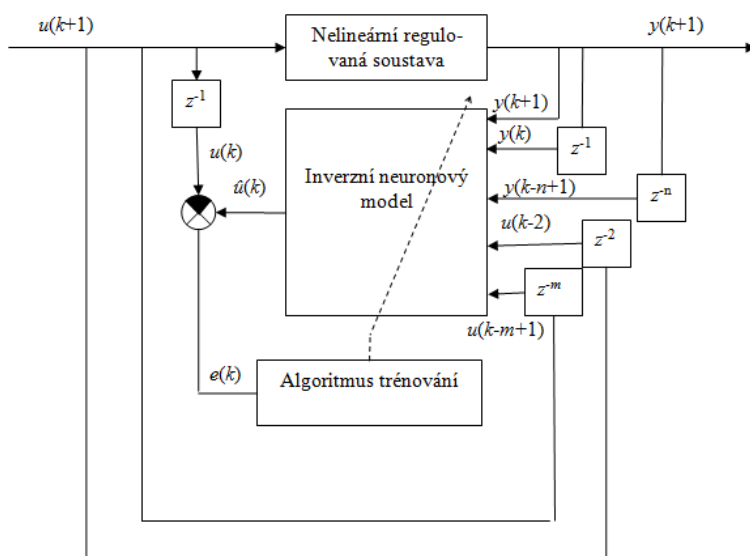
$$u(k) = \varphi^{-1}[y(k+1), y(k), \dots, y(k-n+1), u(k-1), u(k-2), \dots, u(k-m+1)] \quad (5.3)$$

kde  $u$  je akční veličina,  $y$  je výstupní veličina,  $k$  je diskretní čas,  $n$  je počet minulých hodnot výstupní veličiny,  $m$  je počet minulých hodnot vstupní veličiny,  $\varphi$  je funkce závislosti výstupní veličiny na vstupní veličině regulované soustavy a  $\varphi^{-1}$  je inverzní funkce.

Ze vztahu (5.3) je zřejmé, že dochází k porušení kauzality výpočtu a nemůžeme model v tomto tvaru použít. Nic ale nebrání tomu, aby byla vytvořena neuronová síť, která bude trénována v režimu off-line [DOLEŽEL, 2008].

### 5.1.1 Identifikace inverzního modelu regulované soustavy

Co se týká návrhu počtu minulých hodnot  $u(k)$  a  $y(k)$  inverzní neuronový model má stejnou strukturu neuronové sítě o počtu vstupů a skrytých neuronů jako dopředný model neuronové sítě. Jediný rozdíl je v tréninkovém schématu pro obecně zadanou SISO soustavu. Proto vycházíme z rovnice (5.3) [DOLEŽEL, 2008].



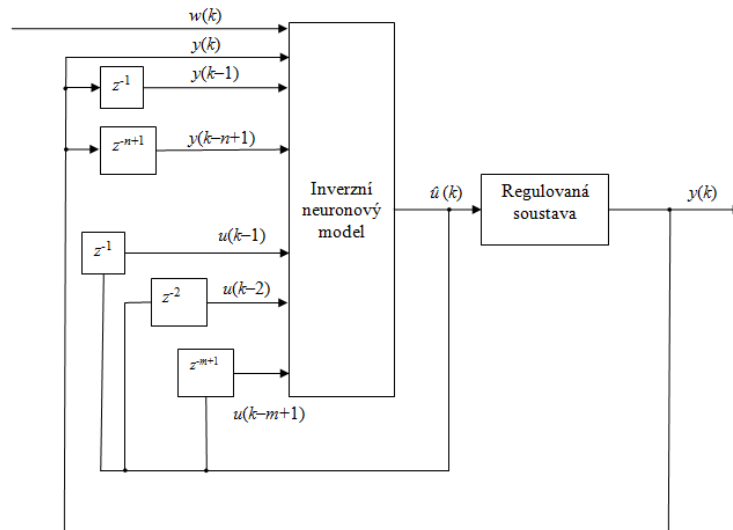
Obr. 37 Trénování inverzního neuronového modelu

### 5.1.2 Popis přímého inverzního řízení

Získáním požadovaného inverzního modelu ve fázi trénování je možné aplikovat a zároveň testovat kvalitu modelu jeho zařazením do regulačního obvodu. Pak je nutné provést nahrazení výstupu  $y(k+1)$  žádanou hodnotou  $w(k)$  [DOLEŽEL, 2008], pak výstup inverzního modelu je dán regresním vektorem

$$\hat{u}(k) = \varphi^{-1} [w(k), y(k), \dots, y(k-n+1), u(k-1), u(k-2), \dots, u(k-m+1)] \quad (5.4)$$

Celý regulační obvod je znázorněn na obr. 38.



Obr. 38 Přímé inverzní řízení s využitím neuronové sítě

Výhodou inverzního řízení je snadná implementace s využitím pouze jednoho neuronového modelu. Nevýhodou je však použití pro systémy s nestabilní inverzí, což je většinou u úloh s vysokou vzorkovací periodou, požadavek pozvolných změn žádané hodnoty, protože musí být splněn požadavek, aby se výstup regulované hodnoty se co nejrychleji blížil žádané hodnotě, nejlépe již během první vzorkovací periody. To není realizovatelné u systémů s velkou setrvačností. Taktéž musí být splněna podmínka stanovení  $T_{vz}$  dle Shanon – Kotelnikova teoremu. Další nevýhodou je horší reakce na poruchy [DOLEŽEL, 2008].

### 5.1.3 Přímé inverzní řízení regulované soustavy se setrvačností 1. řádu

Pro nelineární regulovanou soustavu se setrvačností 1. řádu danou vztahem

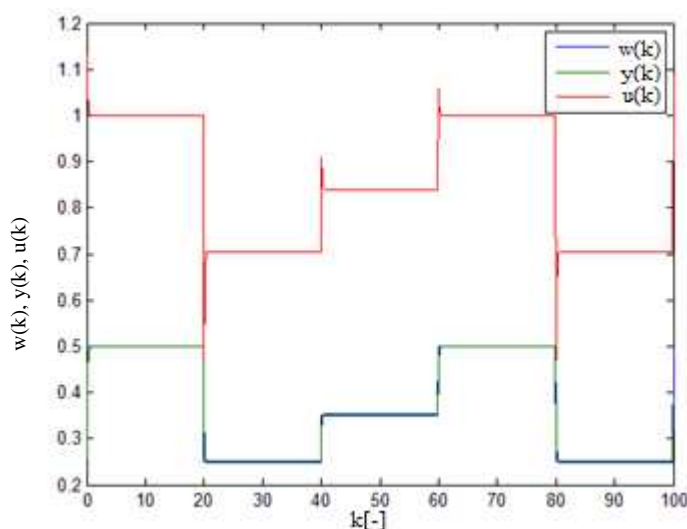
$$y(k+1) - 0.6y(k) = 0.2u^2(k) \quad (5.6)$$

byl navržen a identifikován inverzní neuronový model a aplikován ve struktuře přímého inverzního řízení, kde žádaná hodnota je dána stupňovitou funkcí

$$w(k) = \begin{cases} 0,5 & 0 \leq k \leq 20 \\ 0,25 & 20,1 \leq k \leq 40 \\ 0,35 & 40,1 \leq k \leq 60 \\ 0,5 & 60,1 \leq k \leq 80 \\ 0,25 & 80,1 \leq k \leq 100 \end{cases} \quad (5.7)$$

Průběh žádané hodnoty  $w(k)$ , akční veličiny  $u(k)$  a výstupní veličiny je uveden na obr. 39 pro platný inverzní model identifikovaný a popsáný v kapitole 4.2. Simulační schéma bylo sestaveno v programu Matlab/Simulink a je uvedeno v příloze E.



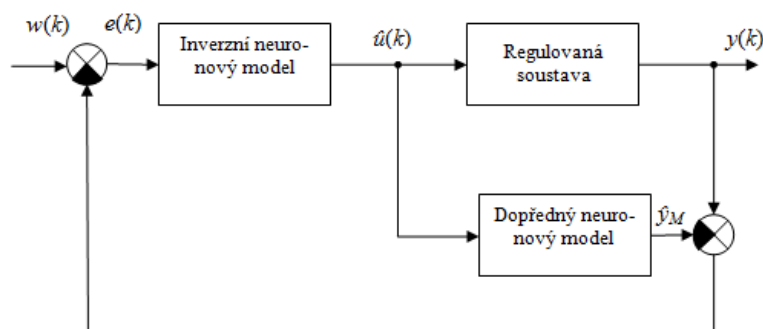


Obr. 39 Přímé inverzní řízení nelineární regulované soustavy se setrvačností 1. řádu

Průběh žádané hodnoty  $w(k)$  a  $y(k)$  se téměř shodují, proto můžeme předpokládat, že inverzní neuronový model pro řízení pomocí neuronových sítí je správně navržen.

## 5.2 Řízení s vnitřním modelem

Metoda řízení s vnitřním modelem (IMC - Internal Model Control) zahrnuje na rozdíl od přímého inverzního řízení také neuronový dopředný model, který je součástí regulačního obvodu (obr. 40).

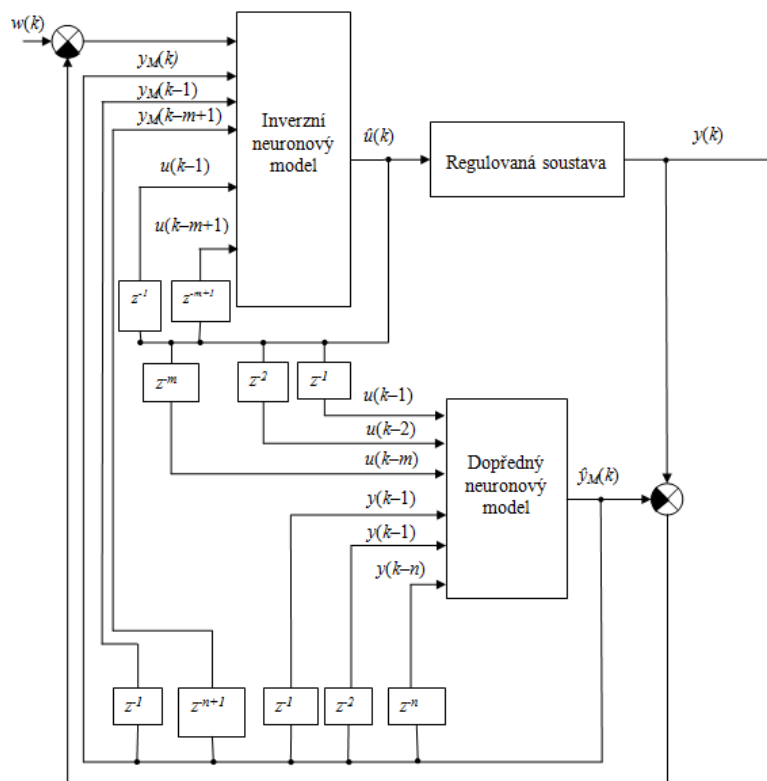


Obr. 40 Řízení s vnitřním modelem využívající neuronových sítí

Řízení s vnitřním modelem kombinuje výhody přímého i zpětnovazebního řízení tzn., že na změnu žádané hodnoty  $w$  rychle reaguje přímá vazba a případné nedokonalosti a poruchy odstraňuje zpětná vazba. Požadujeme kvalitní dopředný model, bez působení poruch. Pokud to není možné, je lepší aplikovat metodu přímého inverzního řízení [DOLEŽEL, 2008].

Základní schéma je potřeba rozvést tak, aby bloky inverzního neuronového regulátoru odpovídaly rovnici (5.5) resp. (5.2). Lze to provést, ovšem za předpokladu, že

existuje porucha na výstupu z regulované soustavy  $y$ , pak se na vstup inverzního neuronového modelu přivádí místo výstupních veličin zatížených poruchou, výstupní hodnoty dopředného neuronového modelu  $\hat{y}_M$  (obr. 41).

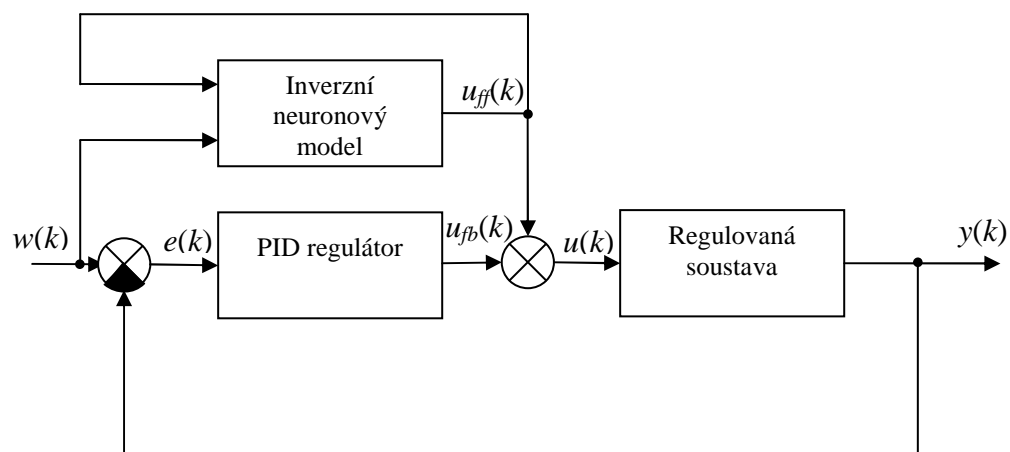


Obr. 41 Podrobné schéma řízení s vnitřním modelem využívající neuronové sítě

Výhody IMC regulace pomocí neuronových sítí jsou obdobné jako u přímého inverzního řízení, ale s působením poruchy je kvalita regulace vyšší. Je nutné si uvědomit, že zpětná vazba umí odstranit chyby způsobené nedokonalým modelem soustavy. Neumí však odstranit chyby způsobené nepřesným inverzním modelem, proto je potřeba provést kvalitní identifikaci obou modelů [DOLEŽEL, 2008].

### 5.3 Dopředné řízení s inverzním modelem

Dopředné řízení s inverzním neuronovým modelem je vhodné tam, kde je požadavek na zkvalitnění stávající PID regulace využitím dopředného neuronového modelu. Metodu je možné využít tam, kde je požadavek na lepší sledování změn žádané hodnoty  $w$ . Princip metody je znázorněn na (obr. 42) [DOLEŽEL, 2008].



Obr. 42 Základní schéma dopředného řízení s inverzním modelem

Za předpokladu, že regulovaná soustava je dána diferenční rovnicí (5.3), pak dopředné složky vstupu neuronové sítě budou sestaveny z minulých hodnot žádané veličiny  $w(k)$  a z minulých hodnot akčních veličin  $u(k)$ . Pak diferenční rovnice bude odpovídat vztahu

$$u_{ff}(k) = \varphi^{-1} [w(k+1), w(k), \dots, w(k-n+1), u_{ff}(k-1), u_{ff}(k-2), \dots, u_{ff}(k-m+1)]. \quad (5.8)$$

Aplikace dopředného řízení s inverzním modelem je vhodné, kde je dopředu známa celá trajektorii žádané veličiny  $w(k)$ . Praktické využití této metody je možné tehdy když je předem k dispozici stabilní PID regulátor. Nevýhodou této metody je, pokud výstup získaného inverzního modelu kmitá [DOLEŽEL, 2008].

## 6 Aplikace neuronových sítí na reálnou soustavu

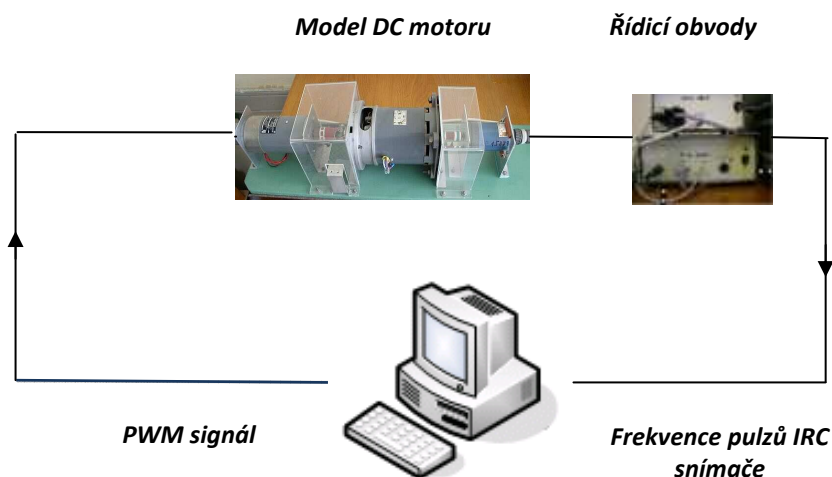
Pro reálné řízení byl vybrán model stejnosměrného motoru, který je umístěn v laboratoři H303. Metoda inverzního řízení a metoda inverzního řízení s vnitřním modelem byly aplikovány na vybraný model z cílem zhodnocení použitelnosti daných metod na reálných soustavách. Obecně je vždy řízení reálného objektu složitější, to se týká i standardních metod řízení. Je to způsobeno především neideálními daty získanými na reálné soustavě. Model stejnosměrného motoru je proporcionální soustava se setrvačností 1. řádu. V další části této práce je popsána jeho identifikace.

Kapitola obsahuje popis navrženého simulačního schématu reálné soustavy, průběhy experimentů, zhodnocení kvality identifikovaného modelu vzhledem k volbě tréninkové množiny dat, typu struktury modelu a počtu vstupů, aktivační funkci a počtu skrytých neuronů daných modelů. Závěr kapitoly obsahuje výsledky regulace stejnosměrného motoru a zhodnocení robustnosti a kvality přímého inverzního řízení a řízení s vnitřním modelem.

### 6.1 Popis laboratorního modelu stejnosměrného motoru

Laboratorní model realizuje stejnosměrný motor, který je sestaven ze stejnosměrného motoru, řídicích a silových obvodů a inkrementálního snímače otáček (IRC-Incremental Rotary Control). K ovládání stejnosměrného motoru z prostředí Matlab/Simulink je k počítači připojena vstupně-výstupní karta MF624 od firmy Humusoft. Motor je ovládán pulsně šířkovou modulací (PWM-Pulse Width Modulation), kdy změnou šířky pulzů se mění napětí na kotvě a tím se mění otáčky motoru. Výstupem je počet pulzů z IRC snímače (Incremental Rotary Control) s 1250 pulzy na otáčku, viz schéma na obr. 43. Motor má vstupní napětí 40 V a vstupní proud 3,3 A.

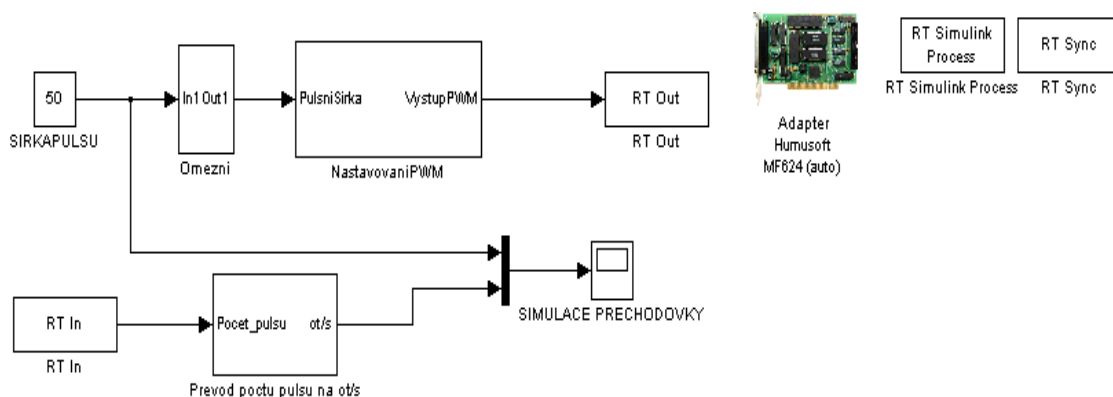
Multifunkční vstupně-výstupní karta MF624 slouží pro připojení počítače k reálné soustavě (k jejím signálům). Multifunkční měřicí karta MF624 je dodávána s ovladači a funkcemi pro daný operační systém. Popis jednotlivých pinů pro připojení reálné soustavy je součástí přílohy L.



Obr. 43 Schématické znázornění úlohy stejnosměrného motoru

## 6.2 Popis simulačního schématu reálné soustavy

Simulační schéma určené k ovládání stejnosměrného motoru je na obr. 44 a bylo vytvořeno v prostředí Matlab-Simulink. Simulační schéma obr. 44 obsahuje podsystém *Nastavování PWM*, v němž se nachází *S-function (PulsniSirka.m)*, která do bloku *Pulse Generator* nastavuje na základě vstupní hodnoty (*PulsniSirka*) šířku pulzu. Tyto vygenerované pulzy jsou pak přivedeny na blok *RT Out*. Tento blok zajišťuje propojení simulačního schématu s reálnou soustavou. Následně je z IRC snímače snímána hodnota pulzů do bloku *RT In* a ta je přivedena na vstup bloku *Prevod pocu pulsu na ot/s* což blok typu *S-function* volající funkci *Citacpulsu.m*. Funkce vyhodnotí počet pulzů za vzorkovací periodu tak, že provede rozdíl signálu v čase  $t$  a signálu v čase  $t-k$ , tj. signál zpožděný o jeden krok simulace. Následně pak signál přivedeme na blok s názvem *Prevod na ot/s*, který je řazen za tímto blokem. Výsledná hodnota je dělena konstantou 1250, což označuje počet pulzů IRC snímače na jednu otáčku a dále vynásobíme konstantou 100, aby odpovídala vzorkovací periodě simulace, tj. 0,01 s. proto že naše vzorkovací perioda výstupní veličiny je 0,01 s. Výpisy algoritmů jednotlivých funkcí a vnitřní struktury bloků jsou součástí přílohy J a K. V tab. 7 je uveden seznam názvů bloků a jejich vzorkovacích period.



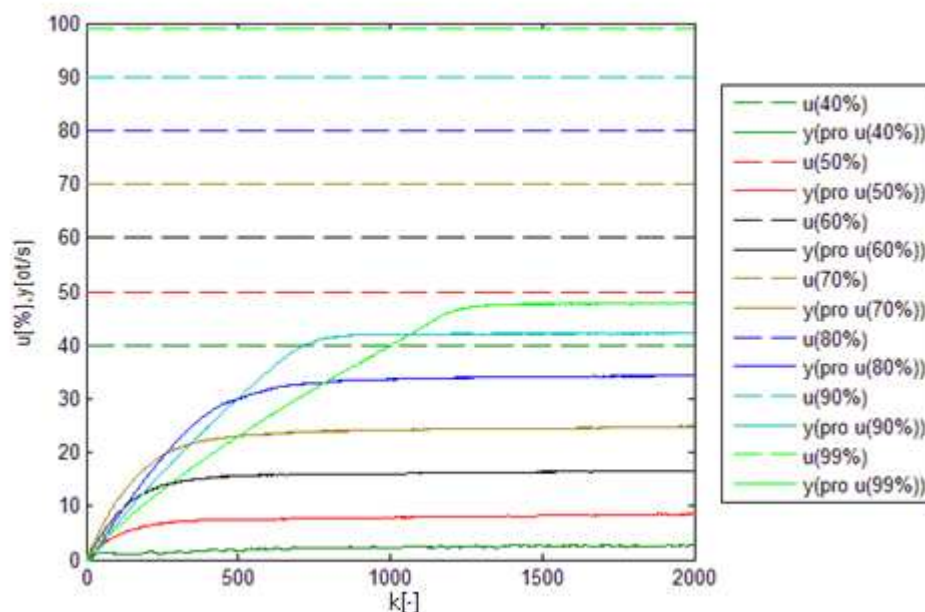
Obr. 44 Simulační schéma pro ovládání stejnosměrného motoru

Tab. 7 Tabulka názvu bloku a příslušné vzorkovací periody

Název bloku	Vzorkovací perioda bloku $T_b$ [s]
Omezení	0,01
PulsniSirka	0,01
Pulse Generator	0,001
RT Out	0,00001
RT In	0,01
CítacPulsu	0,01
Prevod na ot/s	0,01

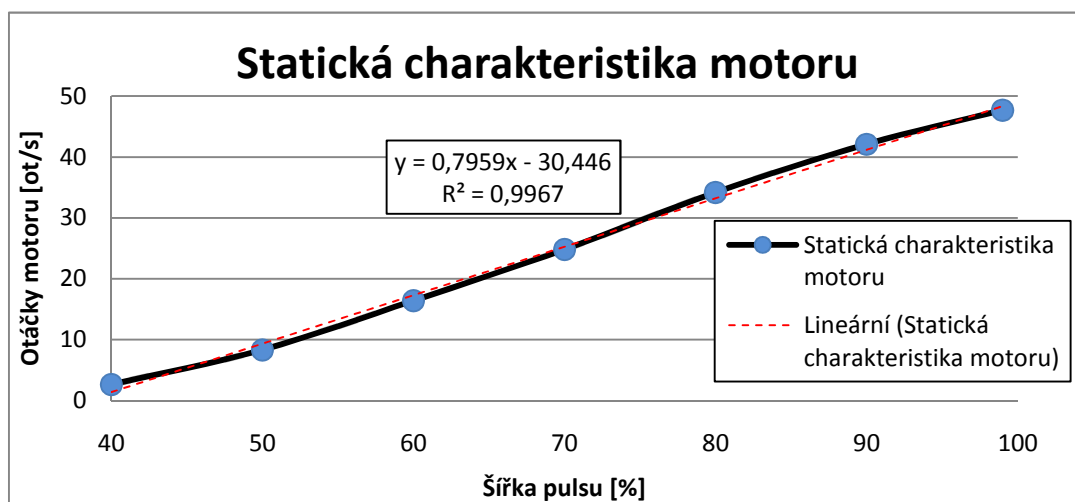
Nižší vzorkovací perioda bloků *Pulse Generator* a *RT Out* by způsobila, že stejnosměrný motor by se začal otáčet od vyšších hodnot šířky pulzu a tím by se nedosáhlo celého rozsahu otáček.

Na obr. 45 jsou přechodové charakteristiky pro různé pracovní oblasti stejnosměrného motoru. Při identifikaci laboratorního modelu stejnosměrného motoru bylo zjištěno, že dynamika této regulované soustavy je různá. Výsledné obrazové přenosy regulované soustavy mají odlišné hodnoty zesílení  $k_p$  i odlišné hodnoty časové konstanty  $T_I$ .



Obr. 45 Přechodové charakteristiky motoru pro různé pracovní oblasti

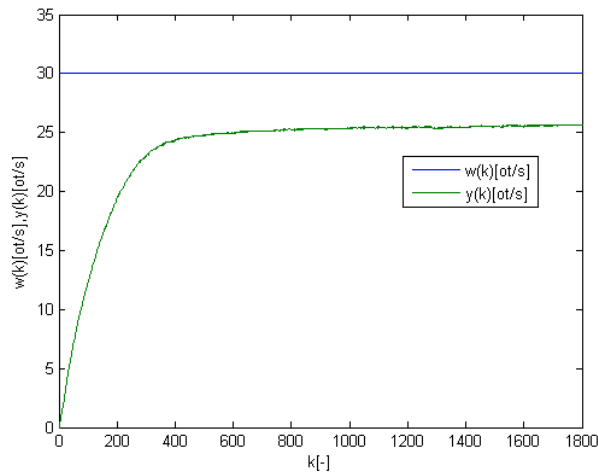
Regulovaná soustava je tedy nelineární, což dokazuje také průběh statické charakteristiky, která byla v práci identifikována (obr. 46).



Obr. 46 Statická charakteristika motoru

Identifikace reálné soustavy pro konkrétní pracovní oblast (obr. 47) je provedena pro simulační ověření metod regulace. Hodnotu střídy (šířky pulsu), kterou nastavujeme do bloku *Pulse Generator*, přepočítáme pomocí rovnice

$$\text{Otáčky motoru} = 0,7959 \cdot \text{Šířka pulsu} - 30,466 \text{ [ot/s]}. \quad (6.1)$$



Obr. 47 Přechodová charakteristika stejnosměrného motoru pro jednu pracovní oblast

Určíme zesílení soustavy ze vztahu

$$k_p = \frac{y(\infty)}{u(\infty)} = \frac{25,44}{30} = 0,848 \quad . \quad (6.2)$$

Identifikace časové konstanty stejnosměrného motoru je dána [VÍTEČKOVÁ, 2008]

$$63,2\% y(\infty) = 18,96 \text{ [ot/s]} \quad (6.3)$$

Vezmeme hodnotu otáček a při vetknutí na osu se vzorky určíme časovou konstantu  $T_I$ .  $T_I$  bude

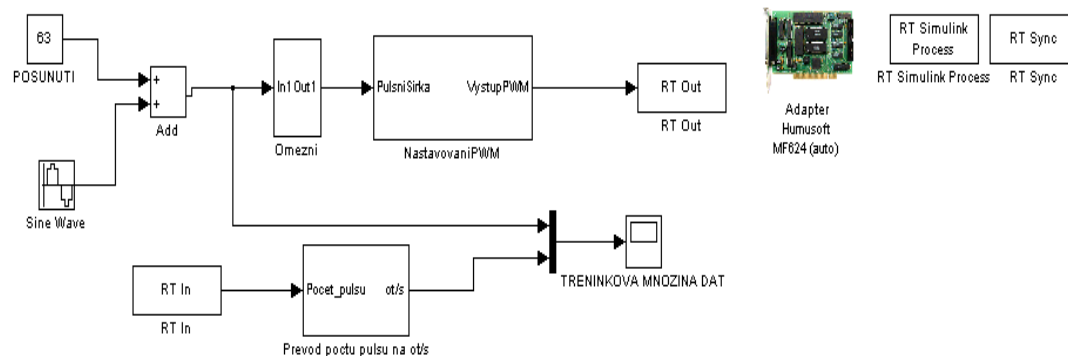
$$T_I = 1,92 \text{ s} .$$

Obrazový přenos soustavy nám určuje vztah (6.4)

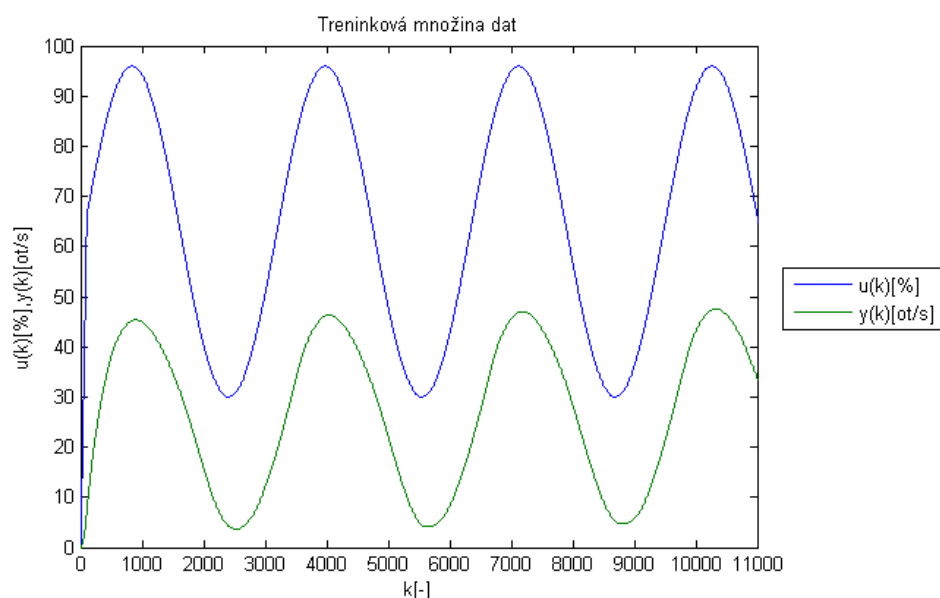
$$G(s) = \frac{k_p}{T_I s + 1} = \frac{0,848}{1,92s + 1} \quad . \quad (6.4)$$

Budícím signálem pro tréninkovou množinu dat byl vybrán sinusový signál s velikostí amplitudy 30 % a frekvencí 0,2 rad/s. Výstupní veličina je zatížena šumem, proto bylo nutné data upravit filtrací, aby se šum, neboli chyba, nepřenášela do procesu trénování neuronového modelu, jehož kvalita na trénovacích datech závisí. Byla provedena filtrace klouzavým průměrem s délkou filtru  $N = 100$ . Vstupní signál množiny dat je na obr. 49. Simulační schéma experimentu na regulované soustavě, jehož výstupem je získání tréninkové množiny dat je uveden na obr. 48.





Obr. 48 Simulační schéma experimentu na stejnosměrném motoru



Obr. 49 Tréninková množina dat stejnosměrného motoru

V následující podkapitole se zaměříme na identifikaci reálné soustavy pomocí neuronových sítí.

### 6.3 Identifikace dopředného neuronového modelu stejnosměrného motoru

Pro dopředné a inverzní modelování je použita architektura dvouvrstvá perceptronová síť s jedním výstupem a aktivační funkce neuronů ve skryté vrstvě je dána hyperbolickou-tangenciální funkcí. Jako strukturu neuronové sítě zvolíme nejčastěji používanou strukturu NARX, která má vektor vstupů sestavený z minulých vstupních i výstupních veličin regulované soustavy a na výstupu dle typu modelování (inverzní, dopředné) je vstupní nebo výstupní veličina regulované soustavy.

Tab. 8 Vlastnosti identifikace reálné soustavy

Dopředné modelování na reálné soustavě	
Architektura sítě	Dvouvrstvá perceptronová síť
Počet neuronů ve skryté vrstvě	3, 6, 10
Aktivační funkce	Hyperbolicko-tangenciální aktivační funkce, Lineární aktivační funkce
Struktura modelu	NARX
Tréninkový algoritmus	Levenbergův-Marquardtův algoritmus
Počet natrénovaných modelů	30
Počet tréninkových epoch	3000
Budící signál tréninkové množiny dat	Sinusový signál
Počet vstupů $m$	2 až 3
Počet výstupů $n$	2 až 6

V tab. 9 až tab. 11 jsou hodnoty mediánu kritériální funkce  $E(2000)$  se změnou počtu neuronů ve skryté vrstvě v závislosti na kombinaci počtu minulých hodnot akční veličiny  $m$  a počtu minulých hodnot výstupní veličiny  $n$  s příslušnou predikcí (5.2). Hodnota mediánu kritériální funkce je ukazatelem pro vhodnou volbu parametrů neuronové sítě a dané struktury modelu NARX s optimálním počtem minulých hodnot.

Tab. 9 Medián hodnot kritériální funkce  $E(2000)$  pro 3 neurony ve skryté vrstvě

$n \backslash m$	2	3	4	5	6
2	$4,1817 \cdot 10^{-8}$	$1,0155 \cdot 10^{-8}$	$9,6484 \cdot 10^{-8}$	$3,1323 \cdot 10^{-7}$	$1,5691 \cdot 10^{-7}$
3		$1,9278 \cdot 10^{-8}$	$1,419 \cdot 10^{-7}$	$8,7051 \cdot 10^{-8}$	$1,5572 \cdot 10^{-7}$

Tab. 10 Medián hodnot kritériální funkce  $E(2000)$  pro 6 neuronů ve skryté vrstvě

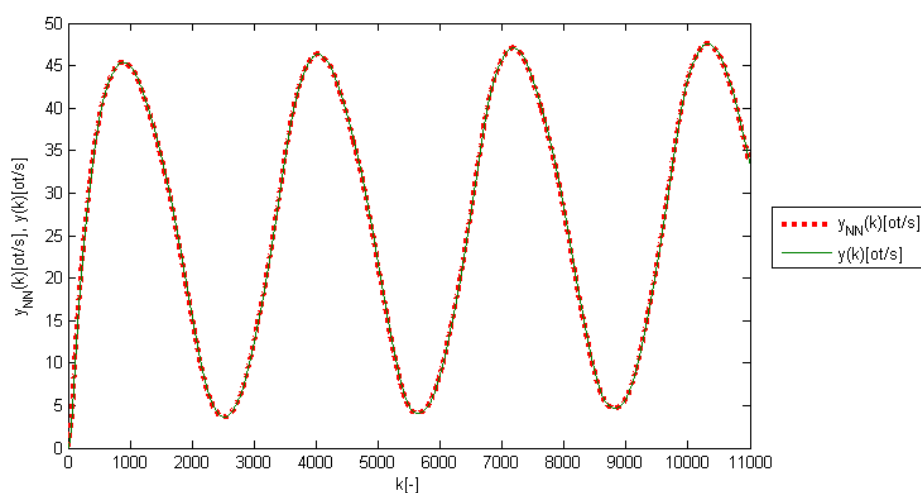
$n \backslash m$	2	3	4	5	6
2	$9,1773 \cdot 10^{-9}$	$1,9193 \cdot 10^{-8}$	$8,7188 \cdot 10^{-9}$	$1,0213 \cdot 10^{-8}$	$2,2978 \cdot 10^{-7}$
3		$6,7617 \cdot 10^{-9}$	$2,2418 \cdot 10^{-7}$	$3,7627 \cdot 10^{-8}$	$8,8822 \cdot 10^{-8}$

Tab. 11 Medián hodnot kritériální funkce  $E(2000)$  pro 10 neuronů ve skryté vrstvě

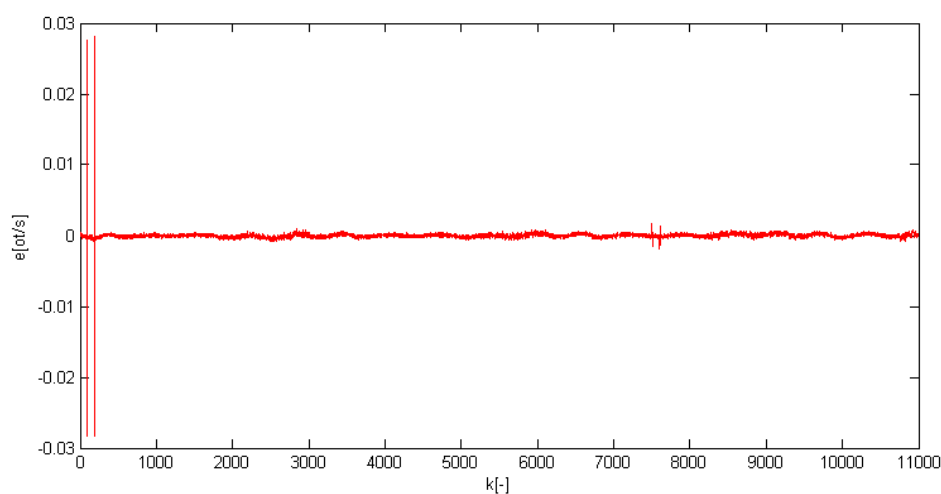
$m \backslash n$	2	3	4	5	6
2	$1,9783 \cdot 10^{-9}$	$4,5415 \cdot 10^{-8}$	$1,121 \cdot 10^{-7}$	$4,4397 \cdot 10^{-9}$	$1,3476 \cdot 10^{-7}$
3		$2,3263 \cdot 10^{-8}$	$4,4383 \cdot 10^{-8}$	$5,1891 \cdot 10^{-8}$	$9,8819 \cdot 10^{-8}$

Z analýzy vhodného počtu parametrů dopředného modelu provedené porovnáním hodnot mediánu kritériální funkce (tab. 9 až tab. 11) byly dále pro testování platnosti dopředného modelu vybrány skupiny modelů, které mají architekturu označenou 5-3-1 (tj.  $m = 2$ ,  $n = 3$  a 3 neurony skryté neurony), dále 6-6-1 (tj.  $m = 3$ ,  $n = 3$  a 6 neuronů skryté neurony) a architektura 4-10-1 (tj.  $m = 2$ ,  $n = 2$  a 10 neuronů skryté neurony).

### Testování neuronové sítě s architekturou 5-3-1



Obr. 50 Predikovaný výstup modelu neuronové sítě s architekturou 5-3-1

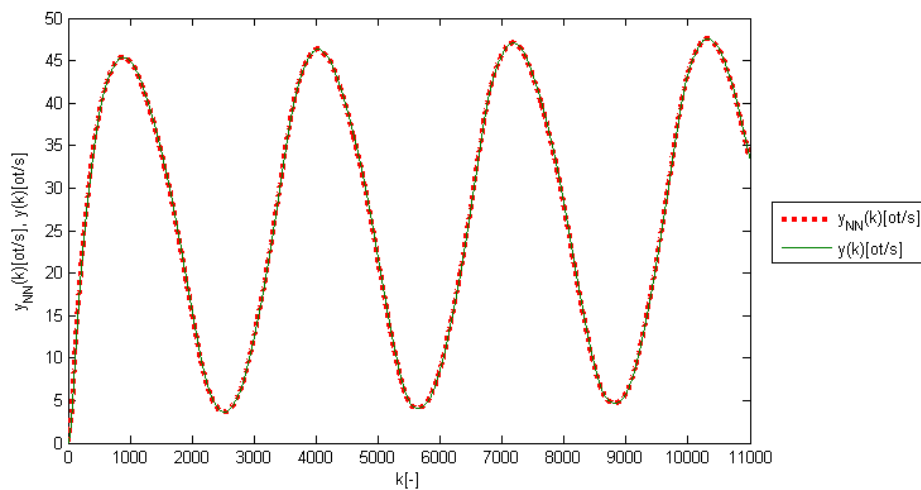


Obr. 51 Průběh chyby identifikace regulované soustavy s architekturou 5-3-1

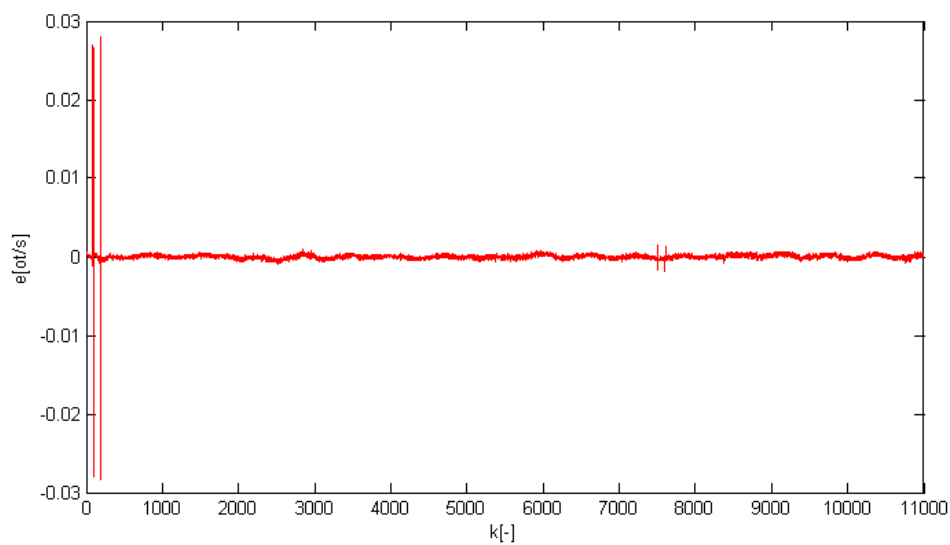
Největší hodnota chyby mezi výstupem regulované soustavy a výstupem dopředného modelu (obr. 51) je pro danou tréninkovou množinu dat dána hodnotou

$$|e_{max}| = 0,0282 \text{ [ot/s]} \quad . \quad (6.5)$$

### Testování neuronové sítě s architekturou 6-6-1



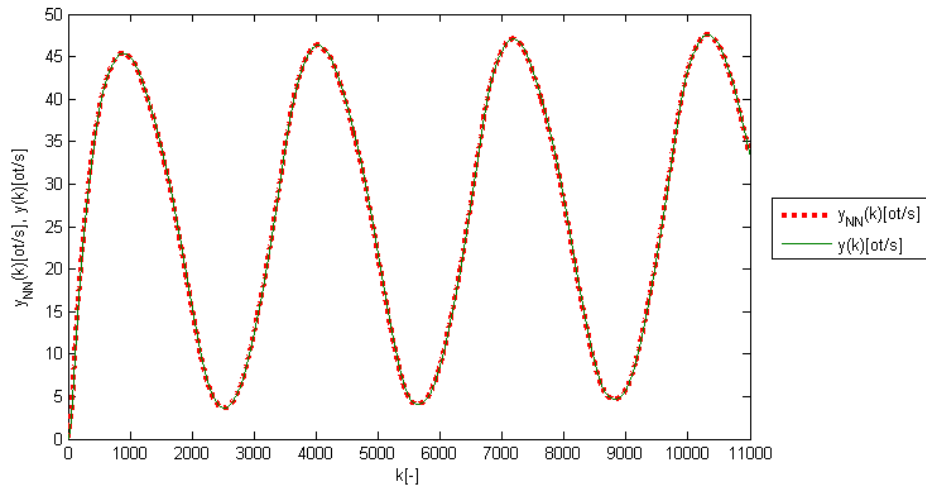
Obr. 52 Predikovaný výstup modelu neuronové sítě s architekturou 6-6-1



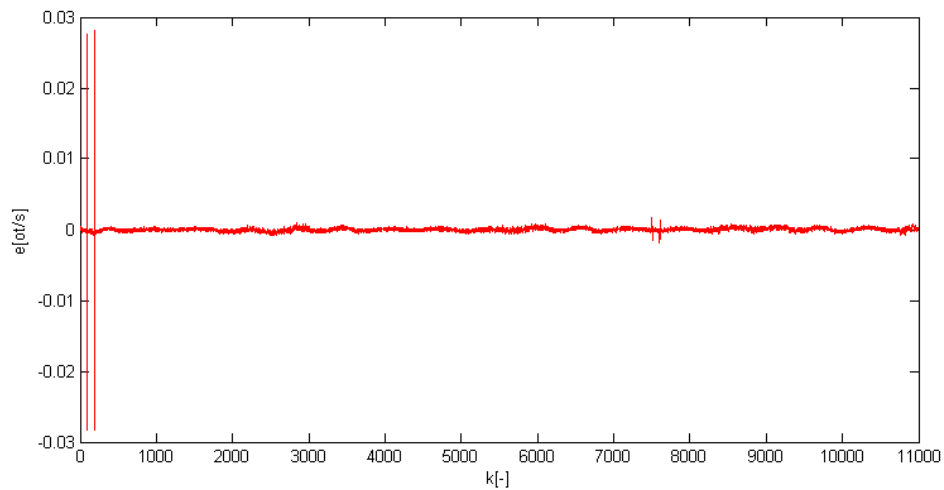
Obr. 53 Průběh chyby identifikace regulované soustavy s architekturou 6-6-1

$$|e_{max}| = 0,0314 \text{ [ot/s]} \quad . \quad (6.6)$$

## Testování neuronové sítě s architekturou 4-10-1



Obr. 54 Predikovaný výstup modelu neuronové sítě s architekturou 4-10-1



Obr. 55 Průběh chyby identifikace regulované soustavy s architekturou 4-10-1

Z obr. 55 určíme největší hodnotu chyby mezi výstupem z regulované soustavy a výstupem z neuronové sítě tzn., že  $|e_{max}|$  bude

$$|e_{max}| = 0,0280 \text{ [ot/s]} \quad . \quad (6.7)$$

Z hodnot v tabulkách tab. 9 až tab. 11 a hodnot  $|e_{max}|$  z rovnic (6.2) až (6.4) nám vyplývá, že se hodnoty mediánu kritériální funkce  $E(2000)$  a  $|e_{max}|$  takřka nemění. Proto jak je popsáno výše, budu volit co nejmenší hodnoty posunutí jak vstupních tak výstupních hodnot proto, aby se případná chyba, která se při řízení neuronovou sítí vyskytne, nešířila sítí ve více krocích. V další podkapitole se podíváme na Přímé inverzní řízení stejnosměrného motoru pomocí neuronových sítí.

## 6.4 Přímé inverzní řízení stejnosměrného motoru

Obecně je metoda přímého inverzního řízení popsána v kapitole 5.1. Simulační schéma přímého inverzního řízení stejnosměrného motoru je součástí přílohy F. Bylo provedeno dopředné modelování s tvorbou několika desítek neuronových sítí s různými počty minulých hodnot akčních veličin, minulých hodnot výstupních veličin a o různých počtech skrytých neuronů. Obecně počet neuronů ve skryté vrstvě je stejný nebo větší, než je počet vstupních hodnot, ale v tomto případě to neplatí, protože při použití architektur sítí v tab. 12 v reálné řízení se stejnosměrný motor vůbec neroztočil, nebo jejich průběh byl dosti kmitavý a nedosáhli bychom jimi ustáleného stavu, tj. požadavky regulace nebyly splněny.

Tab. 12 Použitelnost vybraných topologií v obvodu

Topologie neuronové sítě	$\tilde{E}(2000)$	Použitelnost v obvodu
3-3-1	$2,0523 \cdot 10^{-5}$	5 modelů kmitá kolem $w(k)$
3-5-1	$9,9888 \cdot 10^{-6}$	Nenalezena kvalitní síť
3-7-1	$6,5263 \cdot 10^{-6}$	2 modely kmitají kolem $w(k)$
4-4-1	$1,3301 \cdot 10^{-5}$	Nenalezena kvalitní síť
4-6-1	$1,9736 \cdot 10^{-6}$	Nenalezena kvalitní síť
4-8-1	$1,8524 \cdot 10^{-6}$	Nenalezena kvalitní síť
5-5-1	$2,5450 \cdot 10^{-7}$	Nenalezena kvalitní síť
5-7-1	$2,2780 \cdot 10^{-7}$	Nenalezena kvalitní síť
5-9-1	$2,3908 \cdot 10^{-8}$	Nenalezena kvalitní síť

Zkoušením topologií s nižším počtem neuronů ve skryté vrstvě jsme získali 5 topologií, které jsou víceméně použitelné k řízení této reálné soustavy viz tab. 13.

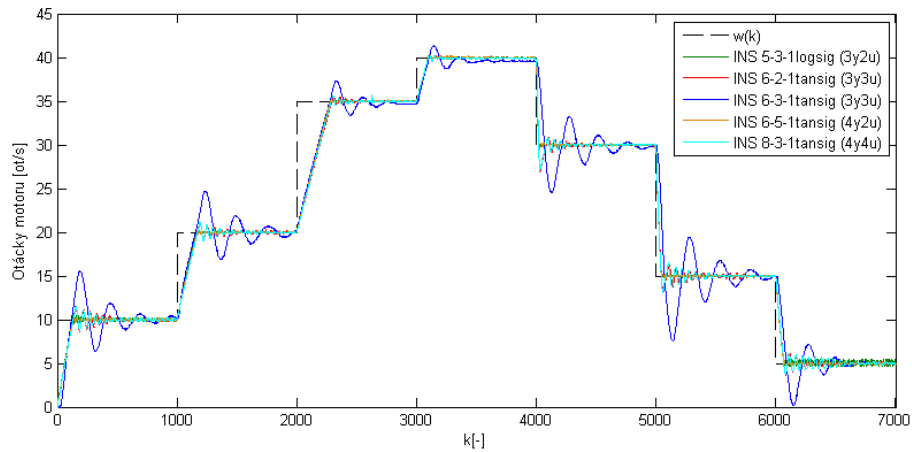
Tab. 13 Vybrané topologie pro metodu přímého inverzního řízení

Topologie neuronové sítě	Rozložení výstupů a vstupů
5-3-1	3y2u
6-2-1	3y3u
6-3-1	3y3u
6-5-1	4y2u
8-3-1	4y4u

Jako testovací signál používám stupňovitou funkci pro žádanou hodnotu  $w(k)$  ve tvaru

$$w(k) = \begin{cases} 10 & 0 \leq k \leq 1000 \\ 20 & 1001 \leq k \leq 2000 \\ 35 & 2001 \leq k \leq 3000 \\ 40 & 3001 \leq k \leq 4000 \\ 30 & 4001 \leq k \leq 5000 \\ 15 & 5001 \leq k \leq 6000 \\ 5 & 6001 \leq k \leq 7000 \end{cases} . \quad (6.8)$$

Průběhy simulace pro jednotlivé topologie jsou k vidění na obr. 56.



Obr. 56 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu Přímého inverzního řízení

Z průběhů na obr. 56 můžeme vyčíst, že pro řízení stejnosměrného motoru se nehodí použití topologie 6-3-1. Tento průběh je kmitavý v celém svém průběhu a nedosáhneme ustáleného stavu. U topologie 8-3-1 můžeme vidět velké kmitání u nízkých otáček, ale při vyšších otáčkách dosahuje výborných hodnot. Abych mohl rozhodnout o kvalitě regulace pro jednotlivé topologie neuronové sítě, použiji tzv. integrální kritérium a to především kritérium IAE (Integral Absolute Error), které je dáno součtem absolutních hodnot regulačních odchylek. Výpočet IAE je následující

$$IAE = \sum_k |e(k)| . \quad (6.9)$$

Tab. 14 Zhodnocení kvality regulace pro metodu přímého inverzního řízení z hlediska kritéria IAE

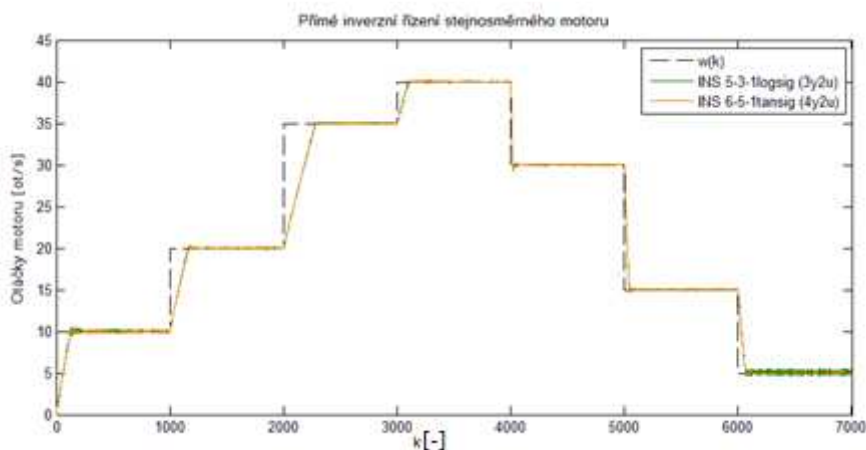
Topologie neuronové sítě	$\sum_k  e(k) $
5-3-1	4856,7
6-2-1	5701,6
6-3-1	11732
6-5-1	4759,8
8-3-1	5729

Z tab. 14 je vidět, že nejvíce vyhovuje topologie 6-5-1 s topologií 5-3-1. Nejméně vyhovuje pak již očekávaně topologie 6-3-1. Nyní zhodnotíme kvalitu regulace z hlediska překmitu  $\kappa_r$  [ot/s] na žádané hodnotě  $w(k)$ .

Tab. 15 Zhodnocení kvality regulace přímého inverzního řízení vyjádřené počátečním překmitem

$w(k)$	5-3-1	6-2-1	6-3-1	6-5-1	8-3-1
	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]
10	0,48	1,52	5,6	0,4	1,52
20	0,24	1,12	4,8	0,32	1,12
35	0,12	0,6	2,36	0,12	0,6
40	0,08	0,4	1,36	0,08	0,4
30	0,63	3,05	5,52	0,8	3,16
15	0,44	2,96	7,55	0,44	2,96
5	0,46	2,32	4,75	0,46	2,37

Uvažujeme-li pásmo regulace  $\pm 3\%$  až  $\pm 5\%$ , pak vyhovují pouze topologie a to 5-3-1 a 6-5-1. Ostatní sítě jsou nevyhovující pro regulaci (nesplňují požadavky kvality regulace). Průběhy regulace pro tyto topologie (5-3-1 a 6-5-1) jsou na obr. 57.



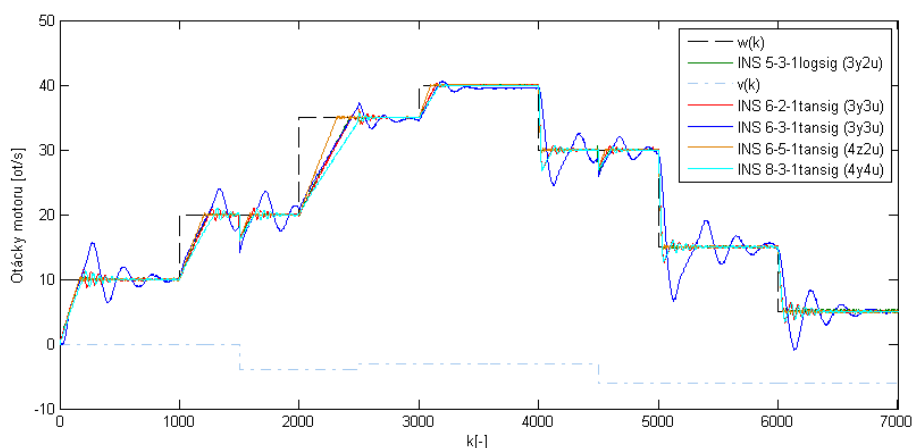
Obr. 57 Průběh regulace vyhovujících topologií pro metodu Přímého inverzního řízení



Přímé inverzní řízení s působením poruchy dané vztahem

$$v(k) = \begin{cases} 0 & 0 \leq k \leq 1500 \\ -4 & 1501 \leq k \leq 2500 \\ -3 & 2501 \leq k \leq 4500 \\ -5 & 4501 \leq k \leq 7000 \end{cases}, \quad (6.10)$$

je další analýzou využití neuronových sítí a jejich aplikovatelnosti v oblasti řízení a konkrétní průběhy jsou zobrazeny v obr. 58.



Obr. 58 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu přímého inverzního řízení s konstantní poruchou působící na výstupu ze soustavy

Z obr. 58 bylo zjištěno, že je porucha nejlépe eliminována neuronovou sítí s architekturou 6-5-1. Nejhorší opět topologie 6-3-1.

Závěrem této podkapitoly si shrneme dosažené výsledky pro přímé inverzní řízení. Pro řízení s inverzním modelem byly získány dva regulátory (5-3-1 a 6-5-1) tvořeny neuronovou sítí s dobrou kvalitou regulace jak pro řízení bez působení poruchy, tak i pro řízení za působení konstantní poruchy  $v(k)$ . A právě schopnost regulátoru odstraňovat působení poruchy bez dalších trvalých regulačních odchylek, je jeden ze základních požadavků řízení.

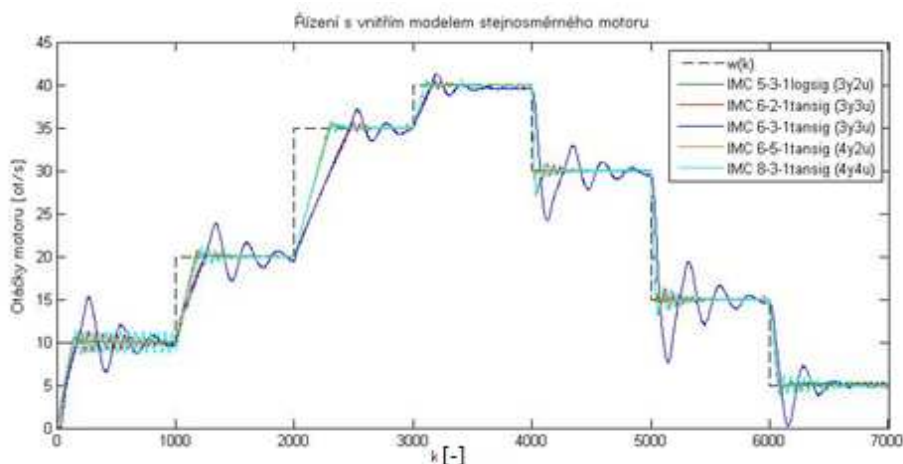
## 6.5 Řízení s vnitřním modelem stejnosměrného motoru

Popis metody je uvedena kapitole 5.2. Simulační schéma této metody je součástí přílohy G. Tato metoda se používá především tam, kde potřebujeme odstraňovat trvalé regulační odchylky. Protože v řízení se žádné trvalé regulační odchylky nevyskytují, není tato metoda příliš vhodná. Nicméně ji vyzkoušíme na stejných architekturách sítí jako u předešlé metody. Pro tuto metodu řízení bylo použito stejných architektur sítí jako u přímého inverzního řízení viz tab. 13. Jako dopředný model soustavy bylo využito sítě typu

5-3-1. U řízení s vnitřním modelem byly sledovány počáteční překmit žádané hodnoty  $w(k)$ . To je způsobeno přivedenými minulými hodnotami odhadu dopředného modelu regulované soustavy. Pro testování této metody řízení byl použit testovací signál jako v kapitole 6.4 daný vztahem

$$w(k) = \begin{cases} 10 & 0 \leq k \leq 1000 \\ 20 & 1001 \leq k \leq 2000 \\ 35 & 2001 \leq k \leq 3000 \\ 40 & 3001 \leq k \leq 4000 \\ 30 & 4001 \leq k \leq 5000 \\ 15 & 5001 \leq k \leq 6000 \\ 5 & 6001 \leq k \leq 7000 \end{cases} . \quad (6.11)$$

Nejprve bylo sledováno, jak se jednotlivé struktury neuronových sítí chovají bez působení poruchy viz obr. 59.



Obr. 59 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu Řízení s vnitřním modelem

Z průběhů je možné vidět, že pro řízení stejnosměrného motoru není vhodné použití struktury neuronové sítě 6-3-1. Tento průběh je kmitavý po celou dobu regulace a požadavek regulace není splněn. Zhodnocení kvality regulace integrálním kritériem IAE pro testovací struktury neuronových sítí je uveden v tab. 16

Tab. 16 Zhodnocení kvality regulace pro metodu řízení s vnitřním modelem z hlediska kritéria IAE

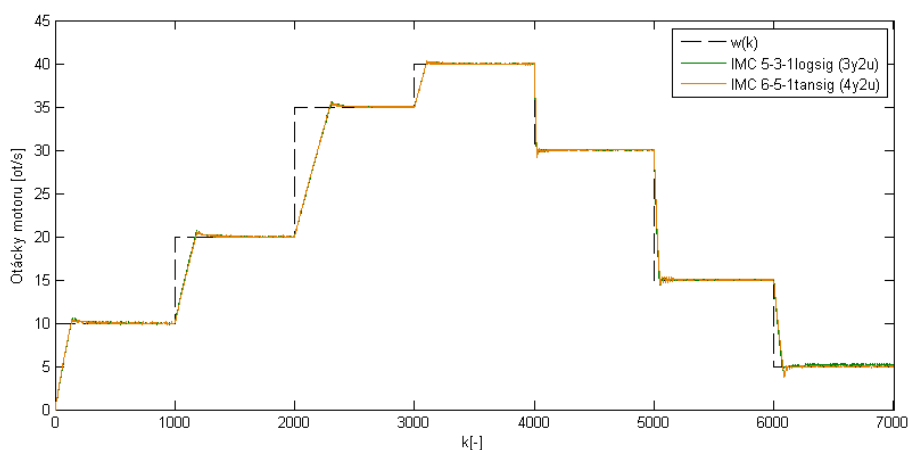
Topologie neuronové sítě	$\sum_k  e(k) $
5-3-1	5050,2
6-2-1	7811,4
6-3-1	13849
6-5-1	5051,2
8-3-1	6613,6

Z tab. 16 je vidět, že nejvíce vyhovují struktury 6-5-1 a 5-3-1. Nejméně pak již očekávaně topologie 6-3-1, což je stejné jako u metody přímého inverzního řízení, nicméně s vyššími hodnotami kritéria IAE. Opět byla zhodnocena kvalita regulace z hlediska překmitu  $\kappa_r$  [ot/s].

Tab. 17 Zhodnocení kvality regulace pro metodu řízení s vnitřním modelem vyjádřené počátečním překmitem

$w(k)$	5-3-1	6-2-1	6-3-1	6-5-1	8-3-1
	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]	$\kappa_r$ [ot/s]
10	0,64	1,44	5,3	0,48	1,52
20	0,72	0,96	3,85	0,64	1,14
35	0,6	0,68	2,12	0,6	0,76
40	0,32	0,48	1,35	0,32	0,56
30	0,8	2,96	5,92	0,8	2,96
15	0,6	2,88	7,4	0,6	2,8
5	0,5	2,24	4,75	2,24	2,24

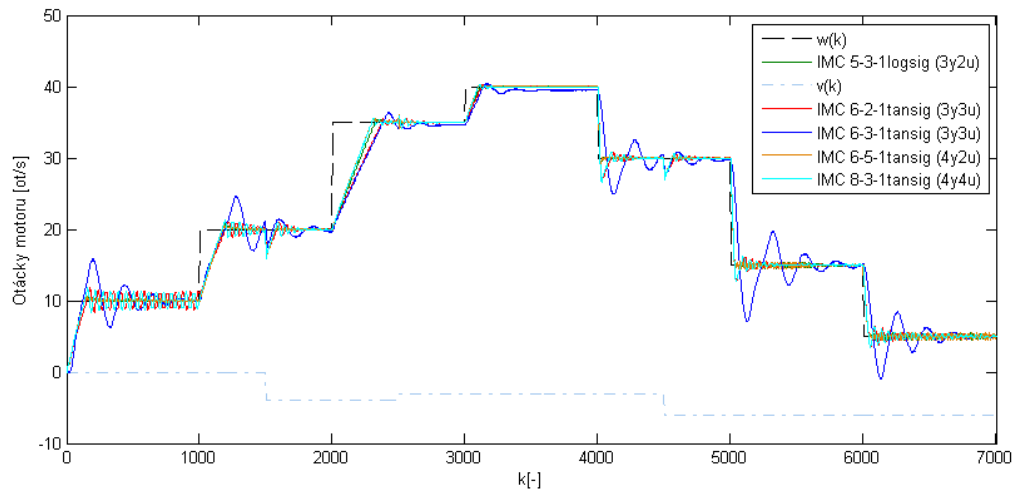
Jak jsem předpokládal, již na začátku dochází k větším překmitům žádané hodnoty  $w(k)$ , což můžeme vidět na obr. 60 pro topologie 5-3-1 a 6-5-1, ale do pásma regulace ( $\pm 3\%$  až  $\pm 5\%$ ) nedostaneme ani jednu z námi vybraných topologií sítí.



Obr. 60 Průběh regulace vyhovujících topologií pro metodu Řízení s vnitřním modelem

Ještě se podíváme na chování jednotlivých sítí při působení konstantní poruchy popsané vzorcem (6.12).

$$v(k) = \begin{cases} 0 & 0 \leq k \leq 1500 \\ -4 & 1501 \leq k \leq 2500 \\ -3 & 2501 \leq k \leq 4500 \\ -5 & 4501 \leq k \leq 7000 \end{cases} \quad (6.12)$$

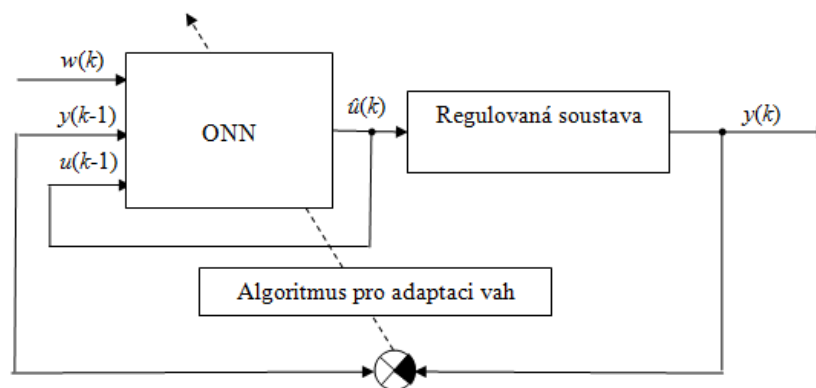


Obr. 61 Průběh řízení na reálné soustavě stejnosměrného motoru pro metodu Řízení s vnitřním modelem s konstantní poruchou působící na výstupu ze soustavy

Při hodnocení kvality regulace zjistíme, že tato metoda řízení se dá použít jen pro vybrané topologie (5-3-1 a 6-5-1), ale musíme předpokládat větší počáteční překymity žádané hodnoty.

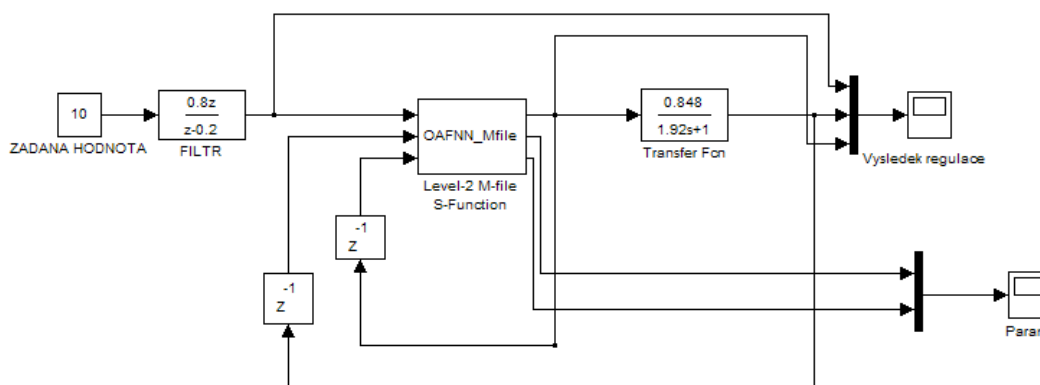
## 6.6 Adaptivní řízení

Adaptivní řízení spočívá v přizpůsobování parametrů regulátoru ke změnám parametrů regulované soustavy tak, aby bylo zajištěno optimální chování systému nezávisle na změnách, které právě nastaly. Adaptivní řízení využívající neuronových sítí vychází z algoritmů adaptivního řízení lineárních soustav, kde popis soustavy a regulátoru je ve tvaru diferenciální rovnice [DRÁBEK, SEIDL, TAUFER, 2006]. Pro simulační a reálné řízení soustavy budu využívat ortogonální síť. Je to výhodné, protože nepotřebujeme získat dopředný ani inverzní model. Algoritmy řízení jsou uvedeny v publikaci [ŠKUTOVÁ, 2009]. Blokové schéma adaptivní regulace za použití ortogonální sítě obr. 62.



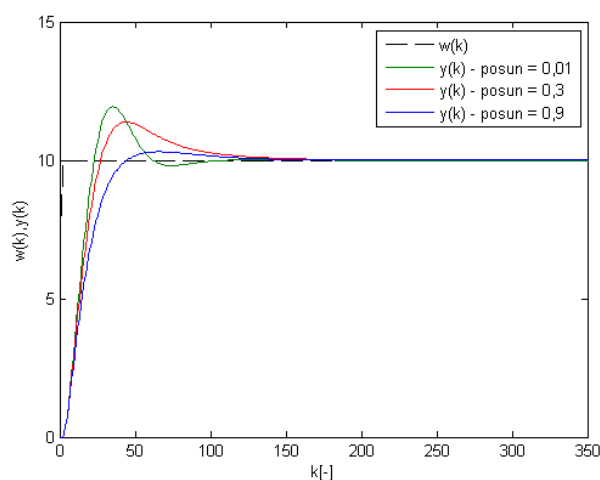
Obr. 62 Blokové schéma adaptivní regulace

Simulační schéma pro adaptivní regulace je na obr. 63. Regulovaná soustava je dána blokem *Transfer Function*, který představuje obrazový přenos regulované soustavy (6.4).



Obr. 63 Simulační schéma adaptivní regulace s využitím ortogonální neuronové sítě pro konkrétní vybraný typ regulované soustavy

Nastavení pro ortogonální neuronovou síť se třemi vstupy ( $w(k)$ ,  $y(k-1)$  a  $u(k-1)$ ) a počet neuronů pro tyto vstupy byl zvolen následovně [5 2 7]. Nyní budeme sledovat změnu parametru *posun* (s proměnným krokem iterace) ze souboru *TrainingOAFNN.m*.

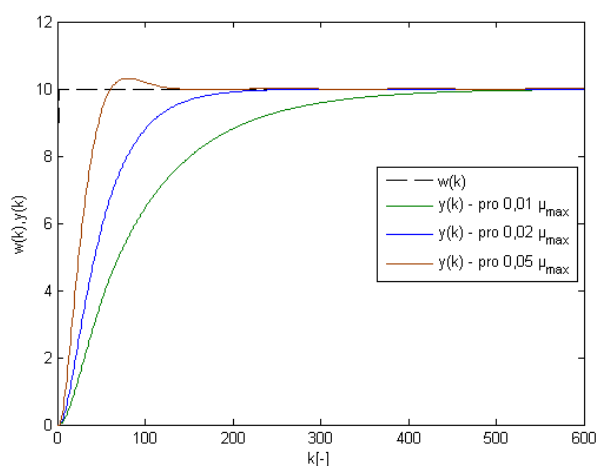
Obr. 64 Adaptivní regulace při změně parametru *posun* pro simulační soustavu

Na obr. 64 je zobrazen průběhy, z nichž lze porovnat vliv zvyšování rychlosti na velikost překmitu žádané veličiny, kdy se zvyšující se parametr posun klesá překmit. můžeme vidět, že při vzrůstající hodnotě parametru *posun* se počáteční překmit žádané hodnoty zmenšuje a kvalita regulace se zvyšuje. viz tab. 18.

Tab. 18 Zhodnocení kvality regulace pro parametr *posun* na simulační soustavě

<b>Krok iterace</b>	<i>0,01posun</i>	<i>0,3posun</i>	<i>0,9posun</i>
<b>IAE</b>	435,66	356,65	156,67
<b><math>\kappa_r</math></b>	1,95	1,37	0,3

Dále bude zhodnocen vliv parametru  $\mu_{\max}$  (konstantní krok iterace) na kvalitu regulace.



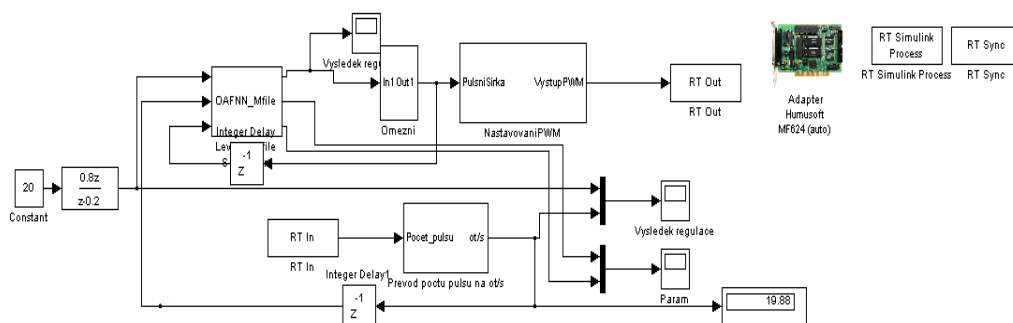
Obr. 65 Adaptivní regulace při změně parametru  $\mu_{\max}$  pro simulační soustavu

Na obr. 65 můžeme vidět, že pokud snižujeme krok iterace  $\mu_{\max}$ , prodlužujeme dobu regulace. Při vyšších hodnotách parametru  $\mu_{\max}$  je doba regulace menší, ale s větším překmitu žádané veličiny. V tab. 19 jsou uvedeny výsledky zhodnocení kvalitu regulace kritériem IAE a zjistíme překmit  $\kappa_r$ .

Tab. 19 Zhodnocení kvality regulace pro parametr  $\mu_{\max}$  na simulační soustavě

<b>Krok iterace</b>	<i>0,01<math>\mu_{\max}</math></i>	<i>0,02<math>\mu_{\max}</math></i>	<i>0,05<math>\mu_{\max}</math></i>
<b>IAE</b>	982,46	528,55	229,67
<b><math>\kappa_r</math></b>	0	0	0,315

Algoritmus adaptivní regulace s ortogonální sítí byl také aplikován na reálnou soustavu. Simulační schéma pro reálné řízení je na následujícím obrázku.



Obr. 66 Simulační schéma adaptivní regulace pomocí ortogonální neuronové sítě pro reálné řízení

Vstupem do bloku *OAFNN\_mfile* je žádaná hodnota  $w$ , a výstupní hodnota (ot/s) zpožděná o jeden krok a akční veličina (pulzní šířka (%)) taktéž zpožděná o jeden krok simulace. Jako testovací signál použijeme stupňovitou funkci ve tvaru

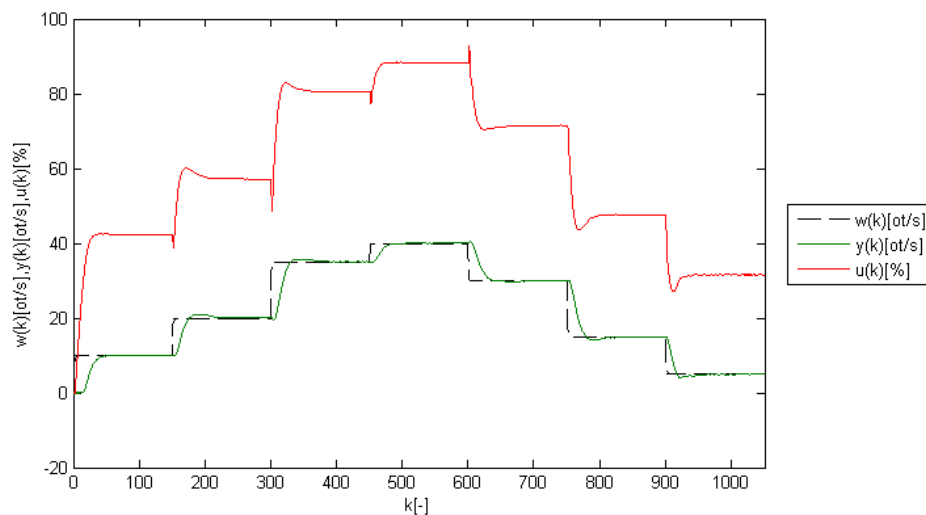
$$w(k) = \begin{cases} 10 & 0 \leq k \leq 150 \\ 20 & 151 \leq k \leq 300 \\ 35 & 301 \leq k \leq 450 \\ 40 & 451 \leq k \leq 600 \\ 30 & 601 \leq k \leq 750 \\ 15 & 751 \leq k \leq 900 \\ 5 & 901 \leq k \leq 1050 \end{cases} . \quad (6.13)$$

Zhodnotíme kvalitu regulace pomocí kritéria IAE. Nastavení pro ortogonální neuronovou síť se třemi vstupy ( $w(k)$ ,  $y(k-1)$  a  $u(k-1)$ ) byl počet neuronů pro tyto jednotlivé vstupy zvolen [5 2 7].

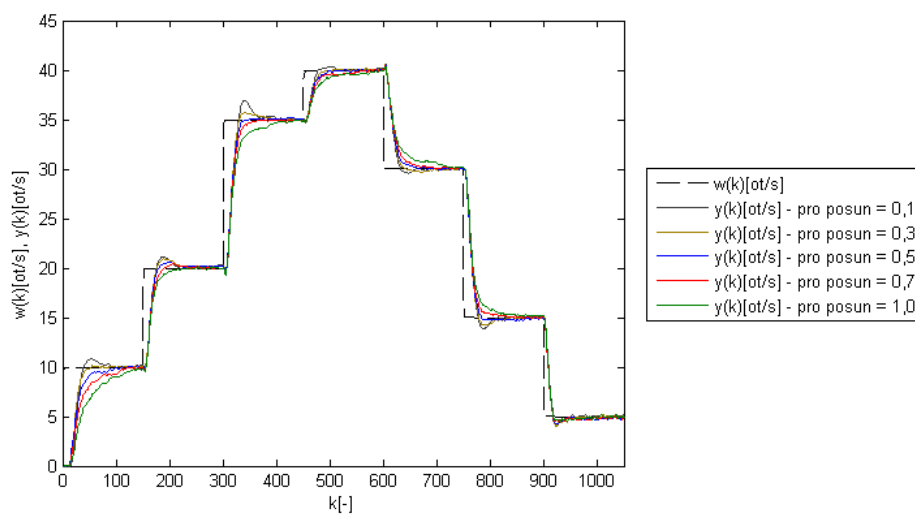
Tab. 20 Zhodnocení kvality Adaptivní regulace pro proměnný parametr *posun*

<i>posun</i>	$\sum_k  e(k) $
0,1	1417,6
0,2	1346,6
<b>0,3</b>	<b>1343,1</b>
0,4	1344,5
0,5	1392,1
0,6	1447,3
0,7	1545,0
0,8	1676,6
0,9	1767,1
1,0	1865,2

Na obr. 67 je průběh regulace s kvalitativně nejlepší hodnotou *posun* tzn. proměnný krok iterace.



Obr. 67 Průběh Adaptivní regulace pro proměnný krok iterace *posun* = 0,3



Obr. 68 Průběhy regulace pro různé hodnoty parametru *posun*

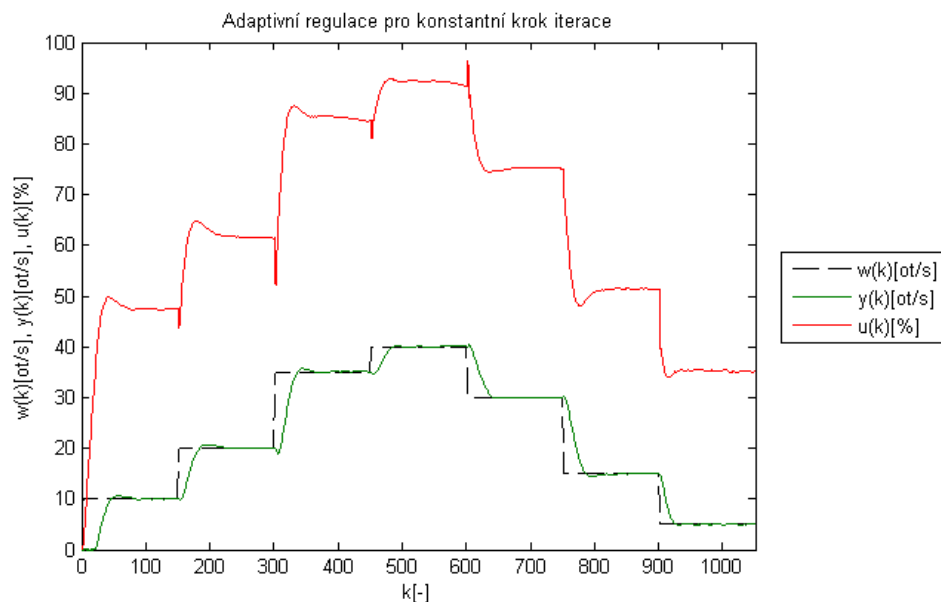
Podle průběhů regulace z obr. 68 bylo zjištěno, že čím je parametr *posun* nastaven na větší hodnotu, tím více se prodlužuje doba regulace. Při nižších hodnotách tohoto parametru dochází k větším překmitům žádané hodnoty  $w(k)$ . Dále byl zhodnocen vliv parametru  $\mu_{max}$ .

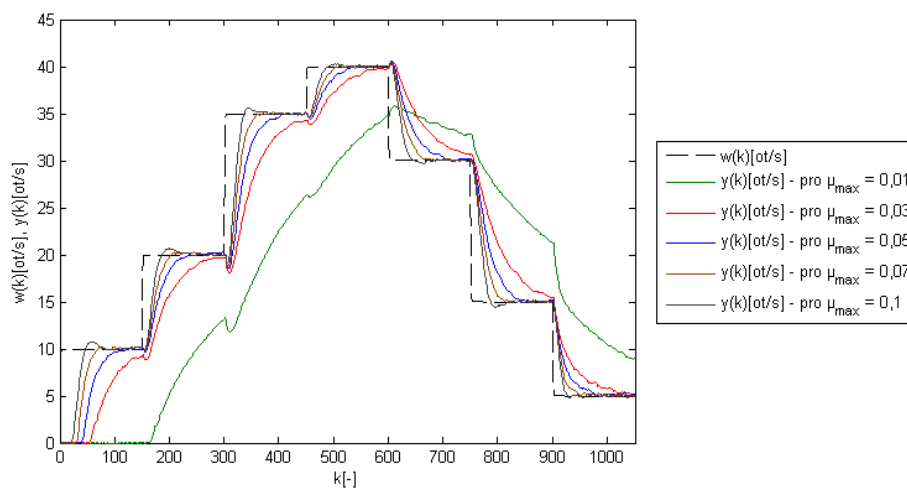


Tab. 21 Zhodnocení kvality Adaptivní regulace pro proměnný parametr  $\mu_{max}$ 

$\mu_{max}$	$\sum_k  e(k) $
0,01	10475,0
0,02	10635,3
0,03	4421,1
0,04	3458,8
0,05	2826,2
0,06	2342,5
0,07	2062,6
0,08	2039,0
0,09	1627,8
0,1	1611,1
0,3	1598,7
0,5	1815,0

Na následujícím obrázku je průběh regulace pro kvalitativně nejlepší hodnoty  $\mu_{max}$  tzn. konstantní krok iterace.

Obr. 69 Průběh Adaptivní regulace pro konstantní krok iterace  $\mu_{max} = 0,3$

Obr. 70 Průběhy regulace pro různé hodnoty parametru  $\mu_{max}$ 

Z průběhů regulace z obr. 70 bylo zjištěno, že pro nízké hodnoty parametru  $\mu_{max}$  dojde k prodloužení doby regulace tzn., že pro nízké hodnoty tohoto parametru nedojde k dosažení žádané hodnoty  $w(k)$ .

## 7 Závěr

Diplomová práce je zaměřená na aplikaci neuronových sítí pro řízení laboratorní úlohy. V úvodu práce jsem se seznámil s architekturou neuronových sítí a jejich aktivačními a agregačními funkcemi. Dále pak s jejich tréninkovými algoritmy a inicializačními metodami. Následně jsem popsal algoritmus identifikace systémů využívající neuronové sítě, tvorbu tréninkové množiny dat a strukturu jednotlivých lineárních modelů využívaných pro identifikaci pomocí neuronových sítí.

Na vybraných typech regulovaných soustav jsem provedl identifikaci dopředného modelu s využitím neuronových sítí. První regulovaná soustava je typu lineární proporcionální soustavu se setrvačností 3. řádu. Pomocí umělých neuronových sítí jsem zjistil nejvhodnější kombinaci vstupů a výstupů a následně pak samotnou strukturu neuronové sítě. Simulaci jsem provedl na dvou signálech a to na signálu stejném jako byl použit při trénování neuronové sítě a na signálu odlišném. Z obou simulací je vidět, že dochází ke shodě mezi výstupním signálem ze soustavy  $y(k)$  a výstupem z neuronové sítě  $y_{NN}(k)$ . Z toho plyne, že námi navržená neuronová síť může nahradit dynamický systém popsany obrazovým přenosem  $G(s)$ . Obdobně jsem postupoval i při identifikaci nelineárního systému popsaného diferenční rovnicí. Opět jsem použil pro testování neuronové sítě dva signály a odezva neuronové sítě na výstup z nelineárního systému byla shodná. Proto jsem postupoval správně.

V další části práce jsem navrhl simulační schéma v prostředí Matlab/Simulink pro řízení reálné soustavy a propojil jsem ho s multifunkční vstupně-výstupní kartou MF624. Následně jsem provedl naměření přechodových charakteristik pro různé pracovní oblasti. Z těchto charakteristik lze vyčíst, že se stejnosměrný motor chová různě pro tyto oblasti. Identifikoval jsem soustavu jako proporcionální soustavu 1. řádu a pro jednu pracovní oblast jsem určil obrazový přenos  $G(s)$ . Dále jsem identifikoval tuto soustavu pro různý počet neuronů ve skryté vrstvě (3, 6, 10). Je vidět, že čím je více neuronů, tím nižší je hodnota mediánu kritériální funkce  $E(N)$ . Po výběru třech nejlepších kombinací vstupů a výstupů s příslušnými predikcemi sítě otestujeme tyto sítě na tréninkové množině dat. Dojdeme k závěru, že mezi nimi není žádný rozdíl. Při realizaci metody inverzního řízení došlo k tomu, že bych nezlískal žádný odpovídající model při dodržení pravidla  $p \geq m + n$ . Proto jsem přistoupil k řešení  $p < m + n$ . Pak po natrénování stovky modelů jsem získal 5 modelů inverzní neuronové sítě, které lze pro řízení „použít“. Při hodnocení kvality regulace pomocí kritéria IAE dostaneme pouze dvě topologie sítě použitelné na řízení při dodržení pásma regulace (5% ze změny žádané hodnoty  $w$ ). Dále jsem se věnoval metodě řízení s vnitřním modelem. Tato metoda se používá především pro soustavy s trvalými

regulačními odchylkami, které se zde nevyskytují. Navíc tato metoda vyžaduje jak kvalitní inverzní model, tak dopředný model neuronové sítě. Počáteční překmity žádané hodnoty jsou větší než u Inverzního řízení, a proto tato metoda je nevhodná pro řízení této reálné soustavy.

Následně mělo být úkolem vyzkoušet adaptivní řízení s inverzním modelem, ale protože jsem nezískal z předchozích kroků kvalitní modely dopředných a inverzních neuronových sítí, aplikoval jsem algoritmy adaptivní regulace využívající ortogonální síť, které byly navrženy vedoucí této diplomové práce. Tyto jsem aplikoval jednak na simulační soustavu, pro kterou jsem stanovil přenos z přechodové charakteristiky naměřené na reálné soustavě a taktéž přímo na reálné soustavě. Provedl jsem zkoumání dvou proměnných parametrů v těchto algoritmech adaptivního řízení jednak pro proměnný krok iterace (*posun*) a jednak pro konstantní krok iterace ( $\mu_{max}$ ). Z těchto průběhů vyplývá, že pro větší hodnoty parametru *posun* dochází k prodlužování doby regulace, ale zase pro nízké hodnoty tohoto parametru dochází k větším překmitům  $w(k)$ . Při snižování parametru  $\mu_{max}$ , dochází k delší době regulace. Jako nejlepší možností řízení stejnosměrného motoru shledávám právě použití adaptivní regulace využívající ortogonální síť. Po zhodnocení kvality regulace jsem dospěl k názoru, že nejlepší možností jak řídit tuto laboratorní úlohu představující stejnosměrný motor je metodou adaptivní regulace při proměnném kroku iterace (*posun*).

## Seznam použité literatury

DOLEŽEL, P. 2008. *Diplomová práce, Umělé neuronové sítě v modelování a řízení kontinuálního bioreaktoru* – Univerzita Pardubice, Fakulta chemicko-technologická, Pardubice 2008

DRÁBEK, O., SEIDL, P., TAUFER, I. 2006. *Umělé neuronové sítě – Základy teorie a aplikace (3)*. [online]. CHEMagazín, Číslo 2, Ročník XVI, 2006 [cit. 2009-04-15]. Dostupný z www: < URL:[http://www.chemagazin.cz/Texty/CHXVI\\_1\\_cl3.pdf](http://www.chemagazin.cz/Texty/CHXVI_1_cl3.pdf) >

DRÁBEK, O., SEIDL, P., TAUFER, I. 2006. *Umělé neuronové sítě – Základy teorie a aplikace (4)*. [online]. CHEMagazín, Číslo 2, Ročník XVI, 2006 [cit. 2009-04-16]. Dostupný z www: < URL:[http://www.chemagazin.cz/Texty/CHXVI\\_2\\_cl9.pdf](http://www.chemagazin.cz/Texty/CHXVI_2_cl9.pdf) >

DRÁBEK, O., SEIDL, P., TAUFER, I. 2006. *Umělé neuronové sítě – Základy teorie a aplikace (14)*. 2006 [cit. 2009-05-16]. CHEMagazín, Univerzita Pardubice, Fakulta elektrotechniky a informatiky, katedra řízení procesů. Dostupný z www: < URL:[http://www.chemagazin.cz/Texty/CHXIX\\_1\\_cl8.pdf](http://www.chemagazin.cz/Texty/CHXIX_1_cl8.pdf) >

FARANA, R. AJ. *Zpracování závěrečných textů z oblasti automatizace a informatiky. Včetně anglicko - českého slovníku automatizační techniky a řízení*. 1.vydání Ostrava: VŠB-TU Ostrava, 2004. 116 s. ISBN 80-248-0557-X.

HUANG S., TAN K. K., TANG K. 2004. *Neural network Control: Theory and Applications*. Baldock: research studies press ldt. 2004. 396 pg.– ISBN 0-96380-285-0

HUMUSOFT S.R.O. *MF 624 Multifiction I/O Card : User Manual* [online]. Dostupné z www: <URL: <http://www2.humusoft.cz/www/datacq/manuals/mf624um.pdf> >

KAJÁN, S. 2006. *Modelovanie a riadenie systémov pomocou neurónových sietí s ortogonálnymi funkciami v prostredí Matlab* [online]. 2006 [cit. 2009-03-20].

Ústav riadenia a priemyselnej informatiky, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave. Dostupný z www:<URL:

[http://dsp.vscht.cz/konference\\_matlab/MATLAB06/prispevky/kajan/kajan.pdf](http://dsp.vscht.cz/konference_matlab/MATLAB06/prispevky/kajan/kajan.pdf)>

NOSKIEVIČ, P. 1999. *Modelování a identifikace systémů*. 1. vyd. Ostrava : MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.

PRINCIPE, J.C., EULIANO N. R., LEFEBVRE W. C. 2000. *Neural and Adaptive Systems: Fundamentals Through Simulation*.- New York : WILEY, 2000. 656 pg. ISBN 0-471-35167-9

ŠKUTOVÁ, J. 2004. *Neuronové sítě v řízení systémů* [online]. VŠB - TU Ostrava, 2004. [cit. 2009-03-20]. Dostupný z www: <URL: <http://www.fs.vsb.cz/books/neuronovesite/index.htm>>.

ŠKUTOVÁ, J. 2008. *Inicializační metody dvouvrstvých dopředných neuronových sítí*. [online]. VŠB - TU Ostrava, 2008. [cit. 2009-04-03]. XXXIII.Seminář ASR 2008, ISBN 978-80-248-1727-9 Dostupný z www: <URL: <http://www.fs.vsb.cz/akce/2008/asr2008/proceedings/307.pdf>>.

ŠKUTOVÁ, J. 2009. *Využití umělé inteligence pro řízení složitých systémů*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2009. 146 stran. Oponenti: Prof. Ing. Ivan Taufer, DrSc., Prof. Ing. Pavel Ošmera, CSc., Prof. Ing. Zora Jančíková, CSc.

TIMKO, J., ŽILKOVÁ J., BALARA D., 2002. *Aplikácie umelých neuronových sietí v elektrických pohonoch*. – Košice, 2002. ISBN 80-85723-27-1

VÍTEČKOVÁ, M., VÍTEČEK, A. 2008. *Základy automatické regulace*., Ostrava: VŠB-TU Ostrava, Fakulta strojní, 2008. 204s. ISBN 978-80-248-1924-2

## Seznam příloh

Příloha A – Seznam obrázků.....	79
Příloha B – Seznam tabulek.....	82
Příloha C – Simulační model pro proporcionální soustavu 3.řádu .....	83
Příloha D – Identifikace modelu nelineární regulovanou soustavu 1. řádu v programu Matlab/Simulink.....	84
Příloha E – Inverzní řízení s využitím neuronových sítí v programu Matlab/Simulink .....	85
Příloha F – Simulační schéma metody přímého inverzního řízení.....	86
Příloha G – Simulační schéma metody řízení s vnitřním modelem .....	87
Příloha H – Simulační schéma metody přímého inverzního řízení s poruchou.....	88
Příloha I – Simulační schéma metody řízení s vnitřním modelem s poruchou .....	89
Příloha J – Algoritmus funkce PulsniSirka.m.....	90
Příloha K – Algoritmus funkce CitacPulsu.m .....	91
Příloha L – Popis pinů karty MF624 pro řízení reálné soustavy .....	92

**Příloha A – Seznam obrázků**

Obr. 1 Model McCullochův-Pittsova neuronu. [DOLEŽEL, 2008] .....	7
Obr. 2 Skoková aktivační funkce .....	10
Obr. 3 Signum aktivační funkce .....	10
Obr. 4 Lineární aktivační funkce .....	11
Obr. 5 Sigmoidální aktivační funkce .....	11
Obr. 6 Hyperbolicko-tangenciální aktivační funkce [ŠKUTOVÁ, 2004] .....	11
Obr. 7 Dvoustvrvá perceptronová síť .....	12
Obr. 8 Radiální neuronová síť [ŠKUTOVÁ, 2004] .....	13
Obr. 9 Hopfieldova síť [ŠKUTOVÁ, 2004] .....	14
Obr. 10 Elmanova síť [ŠKUTOVÁ, 2004] .....	15
Obr. 11 Struktura neuronové sítě s ortogonálními aktivačními funkcemi [KAJAN, 2006] .....	17
Obr. 12 Dvoustvrvá dopředná neuronová síť .....	23
Obr. 13 Procedura identifikace systému s využitím neuronové sítě [ŠKUTOVÁ, 2004] .....	26
Obr. 16 NARMAX struktura modelu [ŠKUTOVÁ, 2004] .....	31
Obr. 17 NOE struktura modelu [ŠKUTOVÁ, 2004] .....	31
Obr. 14 NARX struktura modelu [ŠKUTOVÁ, 2004] .....	31
Obr. 15 NFIR struktura modelu [ŠKUTOVÁ, 2004] .....	31
Obr. 18 Tréninková množina dat .....	33
Obr. 19 Průběh mediánu kriteriální funkce $\tilde{E}(5000)$ trénování dopředného modelu pro proporcionální soustavu 3. řádu .....	34
Obr. 20 Statistické zhodnocení kriteriální funkce $E(5000)$ pro $m = 1$ a různý počet výstupních hodnot $n$ .....	34
Obr. 21 Statistické zhodnocení kriteriální funkce $E(5000)$ pro $m = 2$ a různý počet výstupních hodnot $n$ .....	34
Obr. 22 Statistické zhodnocení kriteriální funkce $E(5000)$ pro $m = 3$ a různý počet výstupních hodnot $n$ .....	34
Obr. 23 Průběh kriteriální funkce $E(N)$ při trénování (učení) pro modely s počtem minulých vstupních signálů $m = 1$ a různým počtem minulých výstupních veličin 35	
Obr. 24 Průběh kriteriální funkce $E(N)$ při trénování (učení) pro modely s počtem minulých vstupních signálů $m = 2$ a různým počtem minulých výstupních veličin 35	
Obr. 25 Průběh kriteriální funkce $E(N)$ při trénování (učení) pro modely s počtem minulých vstupních signálů $m = 3$ a různým počtem minulých výstupních veličin 35	
Obr. 26 Průběh trénování (učení) dopředného modelu regulované soustavy pro různé počty skrytých neuronů .....	36
Obr. 27 Predikovaný výstup modelu neuronové sítě regulované soustavy na tréninkový signál .....	37
Obr. 28 Průběh chyby dopředného modelu testovaného množinou tréninkových dat .....	37
Obr. 29 Predikovaný výstup modelu neuronové sítě regulované soustavy na testovací signál .....	38
Obr. 30 Průběh chyby dopředného modelu testovaného množinou testovacích dat 38	
Obr. 31 Tréninková množina dat pro identifikaci dopředného modelu nelineární regulované soustavy .....	40
Obr. 32 Predikovaný výstup modelu neuronové sítě regulované nelineární soustavy na tréninkový signál .....	42
Obr. 33 Průběh odchylky identifikace modelu regulované soustavy s využitím tréninkové množiny dat .....	42



Obr. 34 Odezva neuronové sítě na testovací signál pro nelineární regulované soustavu .....	43
Obr. 35 Průběh odchylky identifikace nelineární regulované soustavy na testovací množinu dat.....	43
Obr. 36 Regulační obvod Přímého inverzního řízení .....	45
Obr. 37 Trénování inverzního neuronového modelu.....	46
Obr. 38 Přímé inverzní řízení s využitím neuronové sítě .....	47
Obr. 39 Přímé inverzní řízení nelineární regulované soustavy se setrvačností 1. řádu .....	48
Obr. 40 Řízení s vnitřním modelem využívající neuronových sítí .....	48
Obr. 41 Podrobné schéma řízení s vnitřním modelem využívající neuronové sítě ..	49
Obr. 42 Základní schéma dopředného řízení s inverzním modelem .....	50
Obr. 43 Schématické znázornění úlohy stejnosměrného motoru.....	52
Obr. 44 Simulační schéma pro ovládání stejnosměrného motoru .....	53
Obr. 45 Přejchodové charakteristiky motoru pro různé pracovní oblasti .....	54
Obr. 46 Statická charakteristika motoru .....	54
Obr. 47 Přejchodová charakteristika stejnosměrného motoru pro jednu pracovní oblast.....	55
Obr. 48 Simulační schéma experimentu na stejnosměrném motoru .....	56
Obr. 49 Tréninková množina dat stejnosměrného motoru.....	56
Obr. 50 Predikovaný výstup modelu neuronové sítě s architekturou 5-3-1 .....	58
Obr. 51 Průběh chyby identifikace regulované soustavy s architekturou 5-3-1 .....	58
Obr. 52 Predikovaný výstup modelu neuronové sítě s architekturou 6-6-1 .....	59
Obr. 53 Průběh chyby identifikace regulované soustavy s architekturou 6-6-1 .....	59
Obr. 54 Predikovaný výstup modelu neuronové sítě s architekturou 4-10-1 .....	60
Obr. 55 Průběh chyby identifikace regulované soustavy s architekturou 4-10-1 ....	60
Obr. 56 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu Přímého inverzního řízení.....	62
Obr. 57 Průběh regulace vyhovujících topologií pro metodu Přímého inverzního řízení .....	63
Obr. 58 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu přímého inverzního řízení s konstantní poruchou působící na výstupu ze soustavy	64
Obr. 59 Průběh regulace na reálné soustavě stejnosměrného motoru pro metodu Řízení s vnitřním modelem.....	65
Obr. 60 Průběh regulace vyhovujících topologií pro metodu Řízení s vnitřním modelem.....	66
Obr. 61 Průběh řízení na reálné soustavě stejnosměrného motoru pro metodu Řízení s vnitřním modelem s konstantní poruchou působící na výstupu ze soustavy .....	67
Obr. 62 Blokované schéma adaptivní regulace .....	68
Obr. 63 Simulační schéma adaptivní regulace s využitím ortogonální neuronové sítě pro konkrétní vybraný typ regulované soustavy .....	68
Obr. 64 Adaptivní regulace při změně parametru $posun$ pro simulační soustavu ...	68
Obr. 65 Adaptivní regulace při změně parametru $\mu_{max}$ pro simulační soustavu .....	69
Obr. 66 Simulační schéma adaptivní regulace pomocí ortogonální neuronové sítě pro reálné řízení .....	70
Obr. 67 Průběh Adaptivní regulace pro proměnný krok iterace $posun = 0,3$ .....	71
Obr. 68 Průběhy regulace pro různé hodnoty parametru $posun$ .....	71
Obr. 69 Průběh Adaptivní regulace pro konstantní krok iterace $\mu_{max} = 0,3$ .....	72
Obr. 70 Průběhy regulace pro různé hodnoty parametru $\mu_{max}$ .....	73
Obr. 71 Simulační model pro trénování proporcionální soustavy 3.řádu .....	83
Obr. 72 Simulační model pro testování proporcionální soustavy 3.řádu.....	83

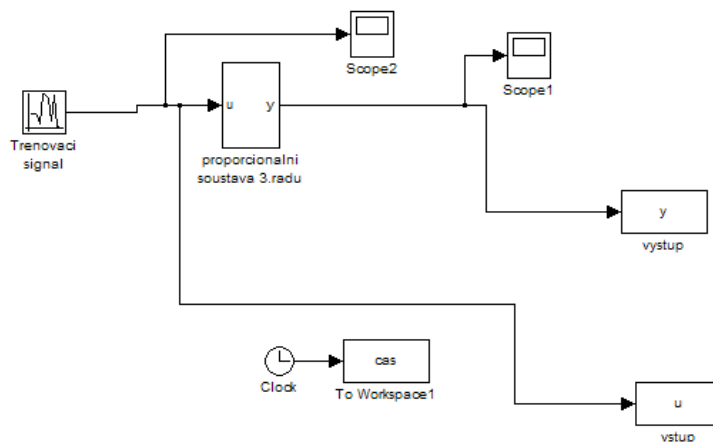
Obr. 73 Simulační model pro trénování neuronového modelu nelineární regulovanou soustavu 1. řádu .....	84
Obr. 74 Simulační model pro testování neuronového modelu nelineární regulovanou soustavu 1. řádu .....	84
Obr. 75 Simulační model inverzního řízení s využitím neuronového modelu .....	85
Obr. 76 Simulační schéma metody přímého inverzního řízení .....	86
Obr. 77 Vnitřní struktura bloku-Omezení.....	86
Obr. 78 Vnitřní struktura bloku - NastavováníPWM .....	86
Obr. 79 Vnitřní struktura bloku - Převod počtu pulsu na ot/s.....	86
Obr. 80 Simulační schéma metody řízení s vnitřním modelem.....	87
Obr. 81 Simulační schéma metody přímého inverzního řízení při působení poruchy na výstupu ze soustavy .....	88
Obr. 82 Simulační schéma metody řízení s vnitřním modelem při působením poruchy na výstupu ze soustavy .....	89

**Příloha B – Seznam tabulek**

Tab. 1 Vlastnosti identifikace pro proporcionální soustavu se setrvačností 3. řádu popsaný diferenční rovnicí.....	32
Tab. 2 Medián kritériální funkce $E(5000)$ pro dopředné modelování proporcionální soustavu 3. řádu .....	33
Tab. 3 Hodnota kritériální funkce $E(5000)$ pro daný počet neuronů ve skryté vrstvě .....	36
Tab. 4 Vlastnosti identifikace pro nelineární regulovanou soustavu popsanou diferenční rovnicí .....	39
Tab. 5 Medián kritériální funkce $E(1000)$ pro nelineární regulovanou soustavu se setrvačností 1. řádu .....	41
Tab. 6 Tabulka závislosti hodnoty kritériální funkce trénování na počtu skrytých neuronů .....	41
Tab. 7 Tabulka názvu bloku a příslušné vzorkovací periody .....	53
Tab. 8 Vlastnosti identifikace reálné soustavy .....	57
Tab. 9 Medián hodnot kritériální funkce $E(2000)$ pro 3 neurony ve skryté vrstvě ..	57
Tab. 10 Medián hodnot kritériální funkce $E(2000)$ pro 6 neuronů ve skryté vrstvě ..	57
Tab. 11 Medián hodnot kritériální funkce $E(2000)$ pro 10 neuronů ve skryté vrstvě .....	58
Tab. 12 Použitelnost vybraných topologií v obvodu .....	61
Tab. 13 Vybrané topologie pro metodu přímého inverzního řízení .....	61
Tab. 14 Zhodnocení kvality regulace pro metodu přímého inverzního řízení z hlediska kritéria IAE .....	63
Tab. 15 Zhodnocení kvality regulace přímého inverzního řízení vyjádřené počátečním překmitem .....	63
Tab. 16 Zhodnocení kvality regulace pro metodu řízení s vnitřním modelem z hlediska kritéria IAE .....	66
Tab. 17 Zhodnocení kvality regulace pro metodu řízení s vnitřním modelem vyjádřené počátečním překmitem .....	66
Tab. 18 Zhodnocení kvality regulace pro parametr $\mu_{\max}$ na simulační soustavě... ..	69
Tab. 19 Zhodnocení kvality regulace pro parametr $\mu_{\max}$ na simulační soustavě .....	69
Tab. 20 Zhodnocení kvality Adaptivní regulace pro proměnný parametr $\mu_{\max}$ ....	70
Tab. 21 Zhodnocení kvality Adaptivní regulace pro proměnný parametr $\mu_{\max}$ .....	72
Tab. 22 Zapojení jednotlivých pinů na kartě MF624 pro reálnou úlohu .....	92

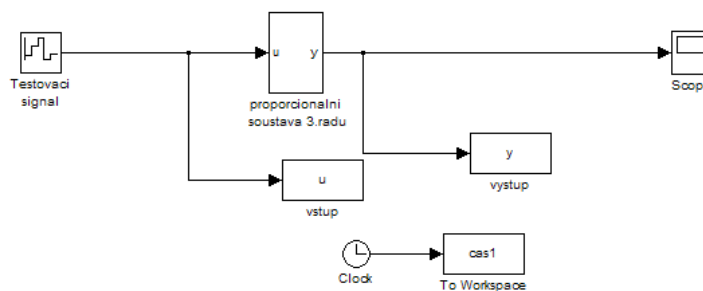
**Příloha C – Simulační model pro proporcionální soustavu 3.řádu**

## Simulační model pro trénování



Obr. 71 Simulační model pro trénování proporcionální soustavy 3.řádu

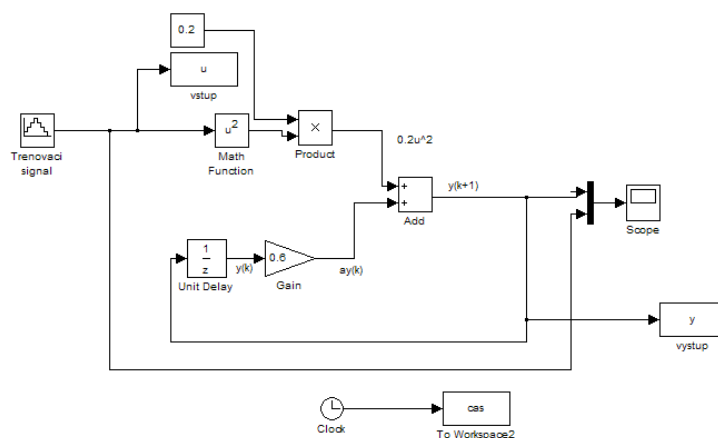
## Simulační model pro testování



Obr. 72 Simulační model pro testování proporcionální soustavy 3.řádu

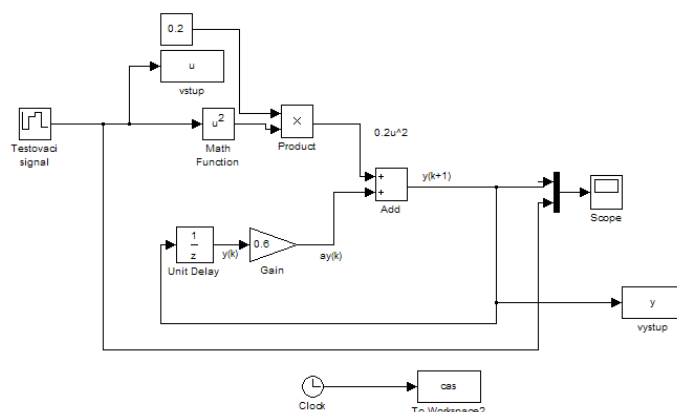
## Příloha D – Identifikace modelu nelineární regulované soustavy 1. řádu v programu Matlab/Simulink

### Simulační model pro trénování neuronového modelu

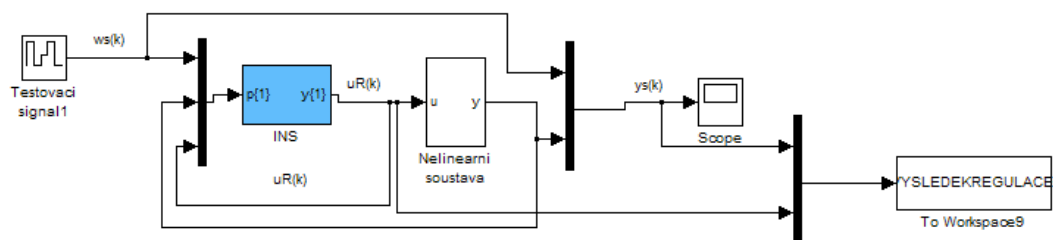


Obr. 73 Simulační model pro trénování neuronového modelu nelineární regulované soustavy 1. řádu

### Simulační model pro testování neuronového modelu

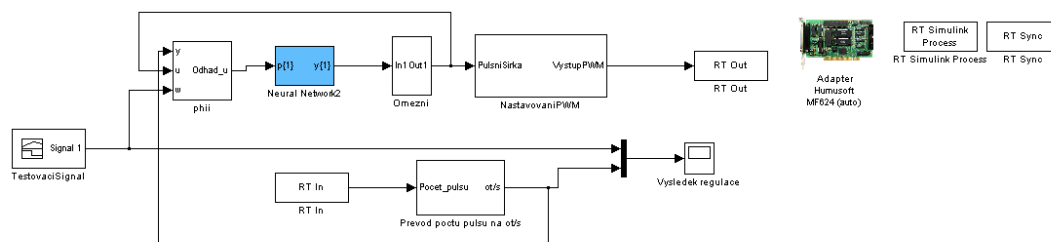


Obr. 74 Simulační model pro testování neuronového modelu nelineární regulované soustavy 1. řádu

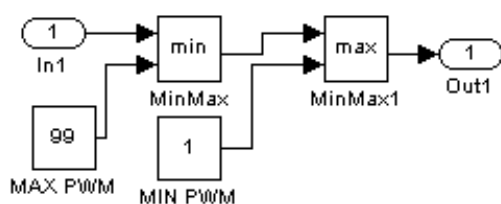
**Příloha E – Inverzní řízení s využitím neuronových sítí v programu Matlab/Simulink**

Obr. 75 Simulační model inverzního řízení s využitím neuronového modelu

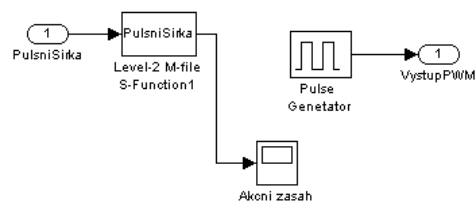
## Příloha F – Simulační schéma metody přímého inverzního řízení



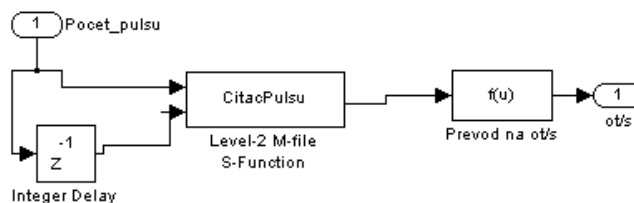
Obr. 76 Simulační schéma metody přímého inverzního řízení



Obr. 77 Vnitřní struktura bloku-Omezeni



Obr. 78 Vnitřní struktura bloku - NastavovaniPWM

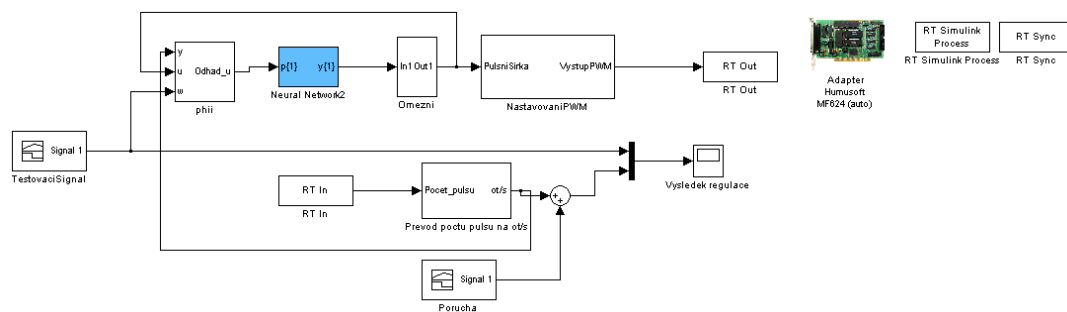


Obr. 79 Vnitřní struktura bloku - Prevod poctu pulsu na ot/s



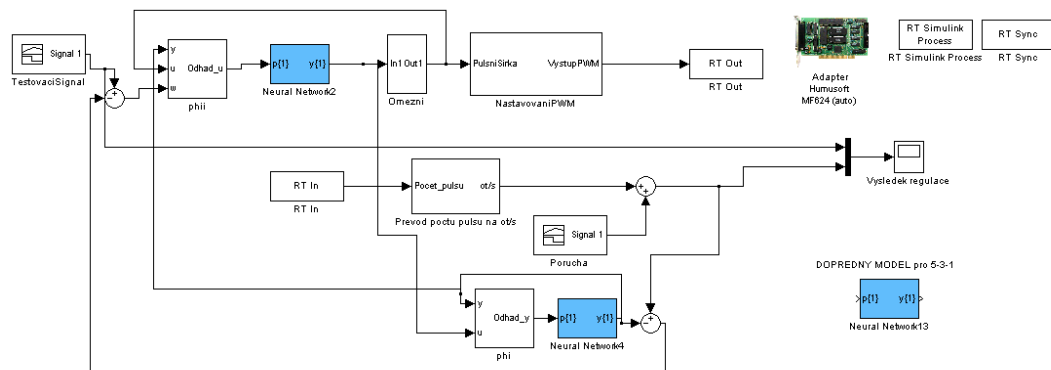


## Příloha H – Simulační schéma metody přímého inverzního řízení s poruchou



Obr. 81 Simulační schéma metody přímého inverzního řízení při působení poruchy na výstupu ze soustavy

## Příloha I – Simulační schéma metody řízení s vnitřním modelem s poruchou



Obr. 82 Simulační schéma metody řízení s vnitřním modelem při působení poruchy na výstupu ze soustavy

**Příloha J – Algoritmus funkce PulsniSirka.m**

```
function PulsniSirka(block)
%%
setup(block);

%%
function setup(block)

    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;

    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;

    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'Real';

    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'Real';

    block.SampleTimes = [0.01 0]; %Vzorkovací perioda bloku

    block.SetAccelRunOnTLC(false);

    block.RegBlockMethod('Start', @Start);
    block.RegBlockMethod('Outputs', @Outputs);
    block.RegBlockMethod('Terminate', @Terminate);

function Start(block)

    block.InputPort(1).Data=0; %Nastaví vstupní port na hodnotu 0

%endfunction

function Outputs(block)

    PulsniSirka=block.InputPort(1).Data; %Přečte hodnotu na vstupní
portu
    block.OutputPort(1).Data= PulsniSirka; %Akční zásah
    %PrepocitanaPulsniSirka=round(((PulsniSirka+13.382)/0.6616));

    set_param([gcs, '/Pulse
Generator'], 'PulseWidth', num2str(round(abs(PulsniSirka))));
    %Nastavení hodnoty PulsniSirka do pole 'Pulse Width' v bloku Pulse
Generator

%endfunction

function Terminate(block)
%endfunction
```

**Příloha K – Algoritmus funkce CitacPulsu.m**

```
function CitacPulsu(block)

setup(block);

function setup(block)

block.NumInputPorts = 2;
block.NumOutputPorts = 1;

block.InputPort(1).Complexity = 'Real';
block.InputPort(1).DataTypeId = 0;
block.InputPort(1).SamplingMode = 'Sample';

block.InputPort(2).Complexity = 'Real';
block.InputPort(2).DataTypeId = 0;
block.InputPort(2).SamplingMode = 'Sample';

block.OutputPort(1).Complexity = 'Real';
block.OutputPort(1).DataTypeId = 0;
block.OutputPort(1).SamplingMode = 'Sample';

block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

block.SampleTimes = [0.01 0]; %Vzorkovací perioda bloku

block.SetAccelRunOnTLC(false);

block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);
block.RegBlockMethod('Terminate', @Terminate);

return

function Start(block)
block.OutputPort(1).Data = 0; %Nastaví vstupní port 1 na hodnotu 0

return

function Outputs(block)
vstuppred = block.InputPort(1).Data; %Vezme hodnotu ze vstupu 1
vstupza = block.InputPort(2).Data; %Vezme hodnotu ze vstupu 2
vysledek = -vstupza + vstuppred; %Odečte od sebe vstup1 a vstup2
block.OutputPort(1).Data = vysledek; %výsledek pošle na výstup

return
function Terminate(block)

return
```

**Příloha L – Popis pinů karty MF624 pro řízení reálné soustavy**

Pro připojení reálné soustavy využijeme oba konektory karty MF624 (X1 i X2)

Tab. 22 Zapojení jednotlivých pinů na kartě MF624 pro reálnou úlohu

<i>Svorkovnice / Pin</i>	<i>Název pinu</i>	<i>Funkce pinu</i>
X1 / 29	GND	Zem
X1 / 30	DOUT0	Výstup generovaného PWM signálu o příslušné střídě
X1 / 31	DOUT1	Výstup generovaného PWM signálu o příslušné střídě
X2 / 1	IRC0A+	Vstup frekvence pulsů z IRC snímače
X2 / 2	IRC0A-	Zem